



# Compression d'images avec et sans perte par la méthode LAR (Locally Adaptive Resolution)

Marie Babel

## ► To cite this version:

Marie Babel. Compression d'images avec et sans perte par la méthode LAR (Locally Adaptive Resolution). Traitement du signal et de l'image [eess.SP]. INSA de Rennes, 2005. Français. NNT : . tel-00131758

**HAL Id: tel-00131758**

**<https://theses.hal.science/tel-00131758>**

Submitted on 19 Feb 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



N° d'ordre : D05-08

# THÈSE

présentée devant

l'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE RENNES

pour l'obtention du titre de

**Docteur de l'INSA de Rennes**

spécialité : *Traitement du Signal*

par

**Marie BABEL-FOUQUET**

---

## **Compression d'images avec et sans perte par la méthode LAR Locally Adaptive Resolution**

---

soutenue le 22 Septembre 2005

### **COMPOSITION DU JURY**

*Président*

Atilla BASKURT      Professeur - INSA de Lyon, LIRIS

*Rapporteurs*

Murat KUNT      Professeur, Ecole Polytechnique Fédérale de Lausanne

Dominique BARBA      Professeur, Ecole Polytechnique de l'Université de Nantes, IRCCyN

*Examineurs*

Stéphane PATEUX      Ingénieur de Recherche et Développement, France Télécom R&D, Rennes

Olivier DÉFORGES      Professeur - Codirecteur de thèse, INSA de Rennes, IETR

Joseph RONSIN      Professeur - Directeur de thèse, INSA de Rennes, IETR





# Remerciements

Ces remerciements ne sauraient débiter sans rendre hommage à Olivier Déforbes, qui m’a encadrée, encouragée, aiguillée, motivée, fait confiance et surtout supportée ! Par son encadrement réellement exceptionnel, il m’a appris ce qu’était la passion, l’investissement auprès de la communauté scientifique et auprès des étudiants. Il sait associer à merveille pertinence, culture, humour et amitié. Il est, à n’en pas douter, devenu mon idole !

Bien sûr, sans mon Grand Chef Joseph Ronsin, je n’aurais pu mener à bien cette thèse... Il m’a accueilli dans son groupe dès le stage de DEA et a réussi le tour de force de me décider à poursuivre dans ce métier. Qu’il considère cette petite victoire comme le résultat de sa très grande gentillesse, sa disponibilité et son dévouement.

Je remercie sincèrement par ailleurs M. Murat Kunt et M. Dominique Barba, les deux rapporteurs de cette thèse, ainsi que M. Atila Baskurt et M. Stéphane Pateux, membres du jury, qui ont accepté (sans connaître l’ampleur de la tâche...) d’examiner ce travail.

Je ne peux évidemment oublier Sylvain Haese dans ces remerciements : d’abord tuteur de mon monitorat, puis coéquipier hippique hors pair, il a su, lors de nos nombreuses conversations, m’indiquer le chemin d’une démarche pédagogique active efficace.

Je suis par ailleurs redevable à l’ensemble du département Electronique et Informatique Industrielle de l’INSA de Rennes, et tout particulièrement à Véronique Haese-Coat, pour m’avoir fait confiance et permis de contribuer à la formation des futurs ingénieurs INSA.

Sans les membres de l’équipe Architecture de l’IETR, je n’aurais jamais eu l’idée de m’intéresser au monde de l’embarqué. Je souhaite ainsi adresser spécialement mes remerciements à Mickaël Raulet et Jean-François Nezan, qui m’ont éclairée et aidée à la réalisation d’une partie de mes travaux.

De même, sans les échanges avec l’équipe Mojette de l’IRCCyN, jamais je n’aurais osé un jeu de mot aussi évocateur que le ”LAR aux Mojettes”, et réaliser un travail conjoint aussi fructueux : un grand merci à Benoît Parrein, Patrick (yankee) Le Callet, Nicolas Normand et Jean-Pierre Guédon.

Je ne peux que remercier mon père de m’avoir montré la voie dans ce métier passionnant qu’est celui d’enseignant-chercheur. Merci encore à mon pôpa et ma maman d’avoir accepté que je suive les traces de mes ancêtres bretons !

*Enfin, je dédie cette thèse à Julien et Klervi, qui trouveront ici enfin concrétisées les raisons de mes absences répétées...*



# Table des matières

<b>Remerciements</b>	<b>iii</b>
<b>Table des matières</b>	<b>xii</b>
<b>Introduction générale et motivations</b>	<b>1</b>
1 Généralités . . . . .	1
2 Objectifs et motivations . . . . .	2
2.1 Compression avec pertes : intérêt d'une représentation en régions . . . . .	2
2.2 Codage sans perte et progressivité : définition des besoins . . . . .	2
2.3 Sécurisation de la transmission des images . . . . .	3
2.4 Compression des images et systèmes embarqués . . . . .	4
2.5 Compression des séquences d'images . . . . .	4
3 Organisation du document . . . . .	4
<b>I Compression avec pertes d'images couleur - Représentation en régions</b>	<b>7</b>
<b>Chapitre 1 La méthode LAR (Locally Adaptive Resolution) :</b>	
<b>principes généraux</b>	<b>9</b>
1 Le codec LAR simple : principe du codage en deux couches . . . . .	9
2 Le codeur spatial . . . . .	10
2.1 Partitionnement . . . . .	11
2.2 Estimation des valeurs moyennes des blocs . . . . .	12
2.3 Encodage des moyennes par prédiction de type MICD . . . . .	13
2.3.1 Quantification des moyennes des blocs. . . . .	13
2.3.2 Prédiction des valeurs moyennes des luminances. . . . .	14
2.3.3 Prédiction des valeurs moyennes des chrominances. . . . .	15
2.4 Post-traitement . . . . .	15
2.5 Caractéristiques du codeur spatial . . . . .	16
3 Le codeur spectral . . . . .	17
3.1 Énergie des blocs à taille fixe / variable . . . . .	18
3.2 Quantification des coefficients $AC'$ . . . . .	18
4 Résultats codec LAR simple . . . . .	20
5 Conclusion . . . . .	21

<b>Chapitre 2</b>	<b>Représentation en régions auto-extractibles</b>	<b>25</b>
1	Préambule . . . . .	26
1.1	Une représentation en régions : pourquoi ? . . . . .	26
1.2	Principe du codeur LAR associé à la description en régions à coût nul . . . . .	27
2	Méthodes de segmentation en régions par graphes d'adjacence . . . . .	27
2.1	Problématique . . . . .	27
2.2	Graphe d'adjacence . . . . .	28
2.3	Classification hiérarchique et métrique . . . . .	28
3	Méthode de segmentation proposée : description à coût nul . . . . .	29
3.1	Distance pondérée . . . . .	29
3.2	Distances pondérées par la moyenne et le gradient. . . . .	29
3.3	Critère de fusion. . . . .	30
3.4	Hiérarchie indicée . . . . .	30
3.5	Elimination des petites composantes. . . . .	31
3.6	Complexité de la segmentation . . . . .	32
4	Applications de la segmentation . . . . .	32
4.1	Codage par ROI de la texture locale . . . . .	32
4.2	Codage des images de chrominances par régions . . . . .	32
5	Segmentation supervisée par le contrôle de la chrominance . . . . .	34
5.1	Description de l'algorithme . . . . .	35
5.2	Résultats et influence des paramètres de segmentation. . . . .	36
5.3	Codage par régions contraintes par masques . . . . .	37
6	Conclusion . . . . .	38
<b>II</b>	<b>Compression sans perte d'images fixes par la méthode LAR : approches pyramidales</b>	<b>41</b>
<b>Chapitre 3</b>	<b>Décomposition Pyramidale : les différentes approches</b>	<b>43</b>
1	Définitions préalables des notions de base . . . . .	43
1.1	"Embedded Coding" . . . . .	43
1.2	Scalabilité et progressivité . . . . .	44
2	Analyse multirésolution . . . . .	46
2.1	Définition intuitive . . . . .	46
2.2	Cadre théorique . . . . .	46
2.3	Orthogonalité et biorthogonalité . . . . .	48
2.4	Pourquoi une décomposition multirésolution ? . . . . .	48
3	La décomposition en sous-bandes . . . . .	49
3.1	Principes généraux . . . . .	49
3.2	Propriétés de la décomposition . . . . .	50
4	De la décomposition en sous-bandes aux ondelettes . . . . .	50
4.1	Base d'ondelettes. . . . .	51
4.2	Algorithme de Mallat : les clés de sa construction. . . . .	51
4.3	Extension de la construction des ondelettes . . . . .	52

4.3.1	Bases biorthogonales d'ondelettes . . . . .	52
4.3.2	Construction d'une transformée dyadique en ondelettes : algorithme à trous . . . . .	53
4.3.3	Schéma avancé de codage par ondelettes : les paquets d'ondelettes . . . . .	53
4.4	Pourquoi la compression d'images par ondelettes est-elle efficace ? . . . . .	54
5	Multirésolution et pyramide : État de l'art . . . . .	54
5.1	Construction de la pyramide : méthodes à redondance de symboles . . . . .	55
5.1.1	Pyramide des moyennes . . . . .	55
5.1.2	Pyramide gaussienne . . . . .	55
5.1.3	Pyramide laplacienne . . . . .	55
5.1.4	Pyramide d'images résiduelles . . . . .	56
5.2	Pyramides réduites . . . . .	57
5.2.1	Pyramide des sommes réduite . . . . .	57
5.2.2	Pyramide des différences réduite . . . . .	57
5.2.3	Pyramide à Transformée en S . . . . .	57
5.3	Pyramide et ondelettes . . . . .	57
5.3.1	Décomposition classique . . . . .	57
5.3.2	Ondelettes de seconde génération : le Lifting Scheme . . . . .	58
5.4	Construction de la pyramide : les méthodes non linéaires . . . . .	61
5.4.1	Décomposition pyramidale morphologique . . . . .	61
5.4.2	Décomposition pyramidale à l'aide de fractales . . . . .	62
6	Conclusion . . . . .	63

## Chapitre 4 Stratégies de codage efficace des décompositions pyramidales - Etat de l'art 65

1	Stratégies d'encodage progressif des valeurs . . . . .	66
1.1	Une technique préliminaire : l'encodage par plages - <i>Run-Length Coding</i> . . . . .	66
1.2	Codage de plan binaire - <i>bitplane coding</i> . . . . .	66
1.3	Significance Map : Exploitation de l'arborescence de la pyramide d'ondelettes . . . . .	67
1.3.1	Construction des zerotrees - L'algorithme EZW . . . . .	68
1.3.2	SPIHT (Set Partitioning In Hierarchical Trees) . . . . .	68
1.3.3	SPECK (Set Partitioned Embedded Block) . . . . .	69
1.4	Inconvénients des méthodes basées <i>zerotree</i> . . . . .	69
2	Codage conditionnel et contextuel . . . . .	70
2.1	Définition du contexte - Modélisation de contexte . . . . .	70
2.2	Exploitation du contexte pour de la compression des images en sous-bandes . . . . .	70
2.2.1	Zerotrees et information contextuelle - LZC . . . . .	70
2.2.2	Encodage conditionnel - Classification . . . . .	71
3	Pyramides et méthodes prédictives . . . . .	71
3.1	Prédiction dans l'espace transformé . . . . .	72
3.1.1	Un précurseur : l'algorithme S+P . . . . .	72
3.1.2	CREW : une compression efficace ? . . . . .	73
3.1.3	Codage pyramidal prédictif et espaces transformés : un aperçu . . . . .	74
3.2	Prédiction dans le domaine spatial . . . . .	75

4	JPEG2000 ou le développement d'un standard . . . . .	76
4.1	La norme de compression . . . . .	76
4.2	Architecture du codeur JPEG 2000 . . . . .	77
4.2.1	Prétraitement . . . . .	77
4.2.2	Transformées, quantification et encodage par blocs . . . . .	77
4.3	Regroupement par paquets et codage par couches . . . . .	78
4.4	Performances de l'algorithme . . . . .	78
5	Remarques générales - Conclusion . . . . .	79

## Chapitre 5 Approche pyramidale prédictive du LAR :

<b>Le LAR-APP</b>	<b>81</b>
1 Introduction et objectifs . . . . .	81
2 Prédiction par l'algorithme de Wu . . . . .	82
2.1 Motivations. . . . .	82
2.2 Sous-échantillonnage de l'image : application de la première passe. . . . .	82
2.3 Deuxième et troisième passes : exploitation du contexte enrichi. . . . .	83
3 L'approche pyramidale prédictive : principes généraux . . . . .	84
4 Construction de la pyramide <i>Bottom-Up</i> . . . . .	86
4.1 Remarques préliminaires . . . . .	86
4.2 Construction indépendante de la grille : méthode générale . . . . .	86
4.3 Prise en compte du pavage LAR - Encodage de la texture . . . . .	87
4.3.1 Encodage de la texture par le LAR-APP . . . . .	87
4.3.2 Encodage de la texture par le codeur spectral . . . . .	88
5 Descente de la pyramide : exploitation du contexte enrichi . . . . .	89
5.1 Niveau $l_{max}$ de la pyramide . . . . .	89
5.2 Obtention de l'image LAR basse résolution . . . . .	89
6 Descente complète de la pyramide - Redondance réduite . . . . .	91
7 Réduction de l'entropie - Modélisation implicite du contexte . . . . .	91
8 Robustesse du schéma - Influence du paramètre de seuil . . . . .	93
9 Compression sans perte : Résultats et comparaisons . . . . .	93
10 Compression avec pertes : Post-traitement et qualité visuelle . . . . .	94
11 Conclusion sur l'approche pyramidale prédictive du LAR . . . . .	97

## Chapitre 6 Codage LAR pyramidal prédictif par transformation en S 99

1 Le LAR-APP : proposition en vue d'améliorer les résultats de compression. . . . .	99
1.1 Faiblesses du LAR-APP. . . . .	99
1.2 Introduction de la transformée en S dans l'algorithme - Entrelacement . . . . .	100
2 Transformée en S et passes de l'algorithme de Wu : Notations et constatations générales . . . . .	101
2.1 Notations : les coefficients transformés . . . . .	102
2.2 Cas du coefficient de moyenne - Première pyramide en S . . . . .	102
3 Définition des prédicteurs - Image des blocs . . . . .	102
3.1 Première pyramide en S . . . . .	103
3.2 Deuxième pyramide en S - Image des blocs . . . . .	103

3.2.1	Moyenne de la deuxième diagonale - Premier coefficient . . . . .	103
3.2.2	Gradient de la deuxième pyramide en S - Image des blocs . . . . .	103
4	Informations de texture et prédicteurs associés . . . . .	104
4.1	Première pyramide en S - Texture . . . . .	104
4.1.1	Gradient de la première pyramide en S . . . . .	104
4.1.2	Remarque : erreur de prédiction sur un coefficient $z_1^l$ d'un bloc de texture . . . . .	104
4.2	Deuxième pyramide en S - Texture . . . . .	104
5	Compression sans perte : résultats . . . . .	104
5.1	Images naturelles : évaluation globale de la méthode - Comparaison avec l'état de l'art. . . . .	105
5.2	Analyse de l'entropie associée aux différentes passes des niveaux de la pyramide . . . . .	106
5.3	Images médicales . . . . .	107
5.4	Images composites - images aériennes . . . . .	112
6	Conclusion - Interleaved S+P . . . . .	113
<b>Chapitre 7 La transformée RWHT+P</b>		<b>115</b>
1	Transformée RWHT . . . . .	116
1.1	Définition de la RWHT . . . . .	116
1.2	Opérations d'arrondis - Fonction de parité . . . . .	117
1.2.1	Somme paire des coefficients . . . . .	117
1.2.2	Somme impaire des coefficients . . . . .	118
2	Pyramide RWHT . . . . .	119
2.1	Construction indépendante de la grille . . . . .	119
2.2	Performances de la RWHT . . . . .	119
3	Pyramide RWHT et Prédiction . . . . .	120
3.1	Définition générale de la phase de prédiction . . . . .	120
3.2	Coefficients multiplicatifs $\beta_m$ et $\beta_k$ . . . . .	121
3.3	Utilisation de la partition Quadtree . . . . .	122
4	Modélisation ou classification de contexte dans l'espace transformé . . . . .	123
5	Combinaison des deux approches - Résultats . . . . .	125
6	Conclusion . . . . .	127
<b>III Le LAR en télémédecine et sur systèmes embarqués</b>		<b>129</b>
<b>Chapitre 8 Télémédecine :</b>		
<b>transmission prioritaire et sécurisée d'images compressées par le LAR</b>		<b>131</b>
1	Contexte de l'étude : la télémédecine - Analyse corrélée au PACS . . . . .	132
2	Codage conjoint source-canal efficace : les hypothèses . . . . .	134
3	Etude de l'impact d'une erreur binaire sur un flux LAR . . . . .	135
3.1	Erreur commise sur la grille . . . . .	135
3.1.1	Système d'encodage de la grille . . . . .	135
3.1.2	Erreur et effet visuel . . . . .	137
3.1.3	Interprétations et applications possibles . . . . .	137



3.2	Récupération incorrecte de la valeur d'une erreur de prédiction . . . . .	140
3.2.1	Impact théorique d'une erreur . . . . .	140
3.2.2	Observation sur des cas réels . . . . .	142
3.3	Remarques . . . . .	148
4	La transformée Mojette-Dirac . . . . .	148
4.1	Transformée directe . . . . .	148
4.2	Propriétés de la transformée. . . . .	149
4.3	Transformée inverse . . . . .	150
4.3.1	Algorithme de reconstruction . . . . .	150
4.4	Critère de reconstruction et redondance - Propriétés de reconstruction . . .	150
4.5	Description multiple . . . . .	151
5	Le LAR aux Mojettes : codage conjoint source-canal . . . . .	153
5.1	Constitution des paquets IP . . . . .	153
5.2	Protection inégale ou différenciée - Qualité image . . . . .	154
6	Apports de la méthode dans le cadre de la télémedecine . . . . .	155
6.1	Réflexions autour du LAR . . . . .	155
6.2	Intérêt de la Mojette . . . . .	156
7	Conclusion . . . . .	157
<b>Chapitre 9 Systèmes embarqués et méthodologie AAA</b>		<b>159</b>
1	SynDEx et méthodologie AAA . . . . .	160
1.1	Présentation générale . . . . .	160
1.2	L'adéquation . . . . .	160
1.3	Génération de code automatique . . . . .	162
2	Les éléments de description sous SynDEx . . . . .	163
2.1	Graphe d'algorithme . . . . .	163
2.2	Graphe d'architecture . . . . .	165
2.3	Portage de l'application sur architecture hétérogène . . . . .	165
3	Modélisation du codec LAR . . . . .	166
3.1	Objectifs du travail effectué - Contraintes de développement . . . . .	166
3.2	Description du codage source sous SynDEx . . . . .	166
3.3	Validation fonctionnelle sur un seul PC . . . . .	171
4	Environnement multi-processeur . . . . .	172
4.1	Application client-serveur : cible multi-PC . . . . .	172
4.2	Cibles multi-PC + multi DSP . . . . .	175
4.2.1	DSP et PC : comprendre les différences . . . . .	176
4.2.2	Plate-forme matérielle hétérogène multi-composants . . . . .	176
4.2.3	Prototypage - Résultats d'implantation . . . . .	177
5	Conclusion . . . . .	181
<b>IV Compression de séquences d'images par la méthode LAR</b>		<b>183</b>
<b>Chapitre 10 Estimation du mouvement</b>		
<b>et extension de la méthode LAR à la vidéo</b>		<b>185</b>

1	Formation du mouvement - Flot optique . . . . .	186
1.1	Formation de l'image et Mouvement 2D . . . . .	186
1.2	Le flot optique : définition . . . . .	187
1.3	Mouvement apparent et mouvement réel . . . . .	187
1.4	Equation de Contrainte du Flot Optique (ECFO) ou Equation de Contrainte du Mouvement Apparent (ECMA) . . . . .	188
2	Estimation du mouvement : méthodologie générale . . . . .	189
2.1	Représentation du mouvement. . . . .	189
2.1.1	Modèle du mouvement. . . . .	190
2.1.2	Support spatial . . . . .	190
2.2	Critères d'estimation. . . . .	191
2.3	Méthodes d'optimisation. . . . .	192
2.3.1	Recherche exhaustive . . . . .	192
2.3.2	Matching : Recherche non-exhaustive - Algorithmes rapides . . . . .	193
2.3.3	Stratégie séquentielle . . . . .	193
2.3.4	Descente de gradient . . . . .	193
2.3.5	Méthodes multirésolution . . . . .	194
2.3.6	Méthodes hiérarchiques . . . . .	195
2.4	Le sens d'estimation . . . . .	196
3	Standard MPEG/H.264 : les solutions envisagées . . . . .	196
3.1	Principe du codage par compensation de mouvement dans H.264 . . . . .	197
3.2	Block Matching et ses dérivées . . . . .	197
3.2.1	Algorithme simple - BMA . . . . .	197
3.2.2	Le DBMA . . . . .	198
3.2.3	HBMA . . . . .	198
3.3	Estimation sous-pixelique . . . . .	199
4	Les méthodes multirésolution basées régions : un aperçu . . . . .	199
4.1	Références . . . . .	200
4.2	Méthodologie générale . . . . .	200
4.2.1	Hiérarchie et Multirésolution : stratégie de parcours de la représentation pyramidale . . . . .	201
4.2.2	Les étapes de construction du flux vidéo scalable . . . . .	202
4.2.3	Un exemple d'algorithme complet : le VideoQ . . . . .	204
5	État de l'art : conclusion . . . . .	205
6	Le LAR et la vidéo . . . . .	205
6.1	Objectifs de l'extension du LAR à la vidéo . . . . .	205
6.2	Etude de la cohérence temporelle de la segmentation . . . . .	206
6.3	Codage vidéo par le LAR : les méthodes envisagées . . . . .	209
6.3.1	Compensation rudimentaire et codage des chrominances par région . . . . .	209
6.3.2	Compensation du mouvement par région, codage de la luminance par bloc . . . . .	209
6.3.3	Compensation et codage global par région . . . . .	210
7	Conclusion . . . . .	210

<b>Conclusions et perspectives</b>	<b>215</b>
1 Synthèse des recherches effectuées lors du travail de thèse . . . . .	215
1.1 Représentation en régions . . . . .	215
1.2 Codage sans perte des images fixes . . . . .	215
1.2.1 Le LAR-APP . . . . .	216
1.2.2 Interleaved S+P . . . . .	216
1.2.3 Méthode RWHT+P . . . . .	216
1.3 Transmission sécurisée d'images : application à la télémédecine . . . . .	217
1.4 Méthodologie AAA et prototypage rapide . . . . .	217
1.5 Extension du LAR à la vidéo . . . . .	218
2 Perspectives . . . . .	218
2.1 Compression des images fixes avec pertes . . . . .	218
2.2 Compression des images fixes sans perte . . . . .	219
2.3 Télémédecine et sécurisation des données . . . . .	219
2.4 Méthodologie AAA et prototypage rapide . . . . .	219
2.5 Le LAR et la vidéo . . . . .	220
<b>Annexes</b>	<b>221</b>
<b>Annexe I Images médicales testées</b>	<b>223</b>
1 Radiographie conventionnelle . . . . .	223
2 Echographie . . . . .	224
3 Coupes tomographiques . . . . .	224
4 Images par résonance magnétique . . . . .	224
4.1 Angiographies . . . . .	224
4.2 Tête . . . . .	225
4.3 Autres parties du corps . . . . .	225
5 Mammographies . . . . .	226
<b>Annexe II Télémédecine : plate-forme de démonstration</b>	<b>227</b>
1 Manipulation des fichiers et fonctions de visualisation . . . . .	227
2 Compression par le LAR-APP . . . . .	228
3 Perspectives . . . . .	229
<b>Lexique</b>	<b>231</b>
<b>Table des figures</b>	<b>241</b>
<b>Liste des tableaux</b>	<b>244</b>
<b>Bibliographie</b>	<b>245</b>

# Introduction générale et motivations

## 1 Généralités

La compression d'une information vise à réduire le volume de données nécessaire pour la représenter. Dans le contexte de l'image numérique, les algorithmes emploient deux stratégies différentes. D'une part, l'exploitation de la redondance spatiale (et temporelle dans le cas des séquences d'images), due à la corrélation existante entre un pixel et ses voisins. D'autre part, certaines informations contenues dans l'image se révèlent peu pertinentes pour l'œil humain et peuvent par conséquent être éliminées.

Deux types de schémas de codage sont alors possibles conduisant à la compression avec pertes ou la compression sans perte des données. La compression sans perte s'avère être une opération totalement réversible : l'image reconstruite correspond intégralement à l'image originale. Les méthodes de cette nature procèdent uniquement à la réduction de la redondance, fournissant de ce fait des performances de compression limitées. Le codage avec pertes apparaît quant à lui comme une opération irréversible, l'image reconstruite étant dégradée par rapport à l'image originale. Les schémas intègrent non seulement une phase de réduction de la redondance, mais aussi un processus d'extraction de l'information pertinente.

Le traitement d'images est une discipline en continuel développement, du fait simultanément du large spectre des applications proposées, et de la montée en puissance des outils numériques. Ainsi, les recherches ne cessent d'apporter nombre de techniques innovantes intégrant de nouvelles fonctionnalités, répondant aux besoins de ses utilisateurs. La conception d'un algorithme générique de compression capable d'obtenir de bonnes performances quel que soit le support reste alors délicat, du fait de la grande diversité de la nature des informations à traiter (images naturelles, médicales, synthétiques, SAR, documents textes ...).

De plus, la masse des données numériques à traiter - audio, image, vidéo, applications multimedia en général - envahit notre environnement. Dans ce contexte, il est indispensable de créer des outils adaptés à une exploitation rapide de telles quantités d'information, tout en conservant l'efficacité nécessaire du point de vue qualité et débit. La compression de ces signaux est ainsi devenue un thème central de recherche des technologies de communications numériques et de stockage.

Diverses méthodes de compression avec ou sans perte (irréversibles ou réversibles) ont été proposées dans la littérature. Les avancées récentes dans le domaine de la compression avec pertes ont vu tout particulièrement l'émergence de techniques s'appuyant sur la quantification vectorielle, les ondelettes, les réseaux de neurones, le codage par fractales. Les méthodes de codage sans perte s'orientent d'une part elles aussi sur l'exploitation de transformées ondelettes réversibles, associées

ou non à une phase de prédiction, et d'autre part sur l'exploitation d'une modélisation de contexte avancée.

C'est dans le cadre d'une telle compétition pour la compression des images que nous avons établi nos recherches pour la conception d'une nouvelle approche originale.

## 2 Objectifs et motivations

L'objectif principal de nos travaux consiste en l'élaboration d'une méthode unifiée pour un codage avec ou sans perte, et ce pour des supports de nature différente : images fixes et vidéos, images naturelles, médicales, aériennes... La technique s'appuie sur le schéma LAR (Locally Adaptive Resolution), initialement conçu pour l'encodage à bas débits des images en niveaux de gris.

Plusieurs pistes de travail ont été avancées afin de proposer de nouvelles fonctionnalités additionnées au codeur LAR de base. Dans cette section, nous présentons chacun des axes de recherche développés.

### 2.1 Compression avec pertes : intérêt d'une représentation en régions

Dans le domaine de la compression avec pertes, la première génération de codeurs intègre des algorithmes issus de la théorie de l'information : le signal est alors encodé pixel par pixel (moyennant certaines opérations du type décomposition, transformation, quantification). Cependant, pour ce type de méthode, le manque de fonctionnalités et de prise en compte du contenu même de l'image restreint son utilisation.

Les codeurs dits de seconde génération [KIK85, TK96] s'attachent en particulier à décrire l'image sous la forme d'un ensemble de zones partageant les mêmes attributs. Ces caractéristiques sont généralement choisies selon des critères issus de modèles du système visuel humain. C'est dans ce cadre que nous avons introduit une description en régions de l'image, directement fondée sur la représentation non-uniforme propre au LAR. Ainsi, cette description apparaît à coût nul (cas des images en niveaux de gris) ou à coût minimal (prise en compte des composantes de chrominance).

Par ailleurs, il est à noter que cette méthode de segmentation sera exploitée dans le cadre de la réalisation d'un schéma de compression de séquences d'images.

### 2.2 Codage sans perte et progressivité : définition des besoins

Le second axe majeur de notre recherche concerne la description d'un algorithme efficace de compression d'images sans perte, autorisant une représentation intermédiaire progressive des données.

**Compression sans perte : pourquoi ?** L'application de traitements non réversibles (filtrage, rehaussement, quantification...), nécessaires pour l'obtention de forts taux de compression lors d'un codage avec pertes, peut détruire une partie essentielle de l'information, ou encore faire apparaître des artefacts conduisant à une interprétation erronée de l'image. Si la compression avec pertes se révèle être la plus usitée, - pour preuve l'ensemble des travaux réalisés autour des

normes JPEG, Jpeg2000 -, de nombreuses applications restent dans le domaine de la *compression sans perte*. Le caractère réversible des opérations menées sur les images assure une reconstruction à l'identique au décodeur du signal présent à l'entrée du codeur. Un domaine de prédilection de cette activité s'avère naturellement être celui de la médecine, en particulier celui de la radiologie, où l'ensemble des données se doit d'être intégralement conservée.

**Compression et scalabilité : un enjeu stratégique.** De façon générale, la décomposition d'un signal en plusieurs parties, à savoir sa composante principale d'une part et ses éléments subordonnés d'autre part, reste une approche fortement présente dans de nombreuses applications de traitement du signal. Le besoin de progressivité spatiale tient en particulier à l'essor des technologies Internet. En effet la transmission d'un flux correspondant à une image codée peut se concevoir comme suit : une image grossière (basse résolution) apparaît puis se voit progressivement raffinée par l'ajout de détails. Il est ainsi souvent utile d'être capable de décrire, transmettre, stocker ou encore reconstruire un signal à différentes échelles, résolutions ou niveaux de qualité. Pour illustrer cette idée, prenons l'exemple d'une source de données unique desservant à travers un réseau un ensemble d'utilisateurs équipés de manière hétérogène. La progressivité introduite par une décomposition du signal original doit fournir à l'utilisateur le mieux équipé (calculateur puissant, haut niveau de résolution du moniteur, réseau haut débit) une image de haute qualité. Dans le même temps, le système doit pouvoir acheminer une version plus simple de l'image à l'internaute connecté par un modem basique. L'accès aux informations à des niveaux de qualité variés, en fonction de la bande passante ou de capacités du terminal (régulation de débit), apparaît comme un facteur clé du développement d'une application distribuée. Ceci est rendu possible par un encodage multirésolution et progressif des données.

**Travail envisagé.** La conception d'une méthode de compression sans perte scalable en résolution passe notamment par la définition d'une structure pyramidale. Associée à un schéma de prédiction efficace et adaptée aux propriétés de la méthode LAR, la décomposition de la pyramide doit permettre de réduire l'entropie résultante des symboles à transmettre.

Si dans un premier temps, nous proposons une solution opérant dans le domaine spatial, l'intégration ultérieure d'une transformée va nous permettre d'améliorer encore les performances de codage de l'approche pyramidale prédictive du LAR.

Par ailleurs, un état-de-l'art sur les techniques de décompositions pyramidales et les stratégies de codage associées est requis. Pour d'autres thématiques connexes abordées (segmentation notamment), le lecteur pourra se reporter à de récentes thèses et HDR soutenues au laboratoire.

## 2.3 Sécurisation de la transmission des images

La télémédecine implique la transmission d'images médicales sur des réseaux à bas débits du type Internet. Ces images constituent des données sensibles : il est évidemment inconcevable de perdre une information hautement confidentielle comme celle relevant d'un diagnostic médical. De ce fait, des techniques de sécurisation des données peuvent être intégrées à une chaîne de transmission d'images compressées.

Nous avons donc envisagé d'associer la méthode de compression LAR, comme codage source, et la transformée Mojette, en tant que codage canal. Cette transformée permet en particulier la protection des paquets IP. L'objectif des travaux réalisés est de proposer une solution efficace de codage conjoint source-canal : la prise en compte de la notion de hiérarchie des flux transmis, inhérente aux schémas multirésolution du LAR, va constituer un point important de la méthode mise en œuvre.

## 2.4 Compression des images et systèmes embarqués

Un système numérique embarqué se définit classiquement comme un système électronique et informatique autonome contenant une partie "intelligente" interagissant avec l'environnement. Un processeur, généralement présent, exécute une fonction spécifique à l'application visée. De nombreux produits intégrant des fonctionnalités liées au traitement de l'image ont récemment fait leur apparition (en téléphonie mobile notamment). Intégrer une application de compression des images devient par la suite une problématique d'actualité.

La méthodologie AAA (Adéquation Algorithme Architecture) propose des solutions pour implanter de façon automatique une application sur une architecture donnée, à partir de sa seule spécification algorithmique. Des recherches sont menées au laboratoire dans ce cadre par l'équipe Architecture. En collaboration avec elle, nous avons projeté de réaliser l'implantation de nos algorithmes de codage (compression avec et sans perte d'images fixes, codage vidéo) sur des plate-formes hétérogènes multi-composants. L'objectif est de déterminer la faisabilité d'un encodage d'images par les méthodes LAR sous la contrainte temps-réel d'une vidéo (moins de 40 ms pour une image).

## 2.5 Compression des séquences d'images

La dernière problématique de cette thèse concerne l'extension de la méthode LAR au cas de la vidéo. Au même titre que la compression des images fixes avec pertes, le codage des séquences d'images peut tirer un grand avantage d'une représentation en régions. Ainsi, le principe d'une représentation en objets fait désormais partie intégrante de MPEG-4 [MPEG-4 Committee00] pour la compression et la manipulation de flux vidéo, et apparaît comme un pré-requis évident pour des problématiques d'indexation (MPEG-7 [Gro99]).

Notre algorithme de segmentation, initialement élaboré pour le traitement des images fixes, montre des performances visuelles intéressantes au regard du faible coût engendré. En intégrant cette approche dans le cadre de la vidéo, nous pouvons espérer concevoir une approche orientée régions essentiellement efficace à très bas débits, et offrant plusieurs niveaux de progressivité.

## 3 Organisation du document

Cette thèse, s'appuyant sur l'approche LAR, se divise en quatre parties distinctes, à savoir : compression avec pertes, compression sans perte, applications aux domaines de la télémédecine

et de l'embarqué et enfin compression vidéo. Ces quatre entités sont elles-mêmes subdivisées en chapitres présentés ci-après.

Tout d'abord, nous exposons au chapitre 1 les principes fondamentaux de la méthode LAR, dans le cadre de la compression avec pertes des images en niveaux de gris et en couleur. Le chapitre 2 introduit l'algorithme de segmentation des images couleur propre au LAR : nous y évoquons en particulier les applications avancées du type codage par région d'intérêt.

La seconde partie, relevant de la compression sans perte, débute par un état de l'art des schémas fondés sur la construction d'une pyramide (chapitre 3), et des stratégies de codage associées (chapitre 4). Par la suite, trois algorithmes de compression sans perte, exploitant chacun une structure pyramidale et les propriétés intrinsèques du LAR, sont proposés. Le premier, dénommé LAR-APP (Approche Pyramidale Prédictive du LAR), intègre un mécanisme efficace de prédiction opérant dans le domaine spatial (chapitre 5). Le chapitre 6 présente une solution dérivée de la première : le processus de codage, l'Interleaved S+P, s'exécute cette fois-ci dans le domaine transformé via une application originale de la transformée en S. Enfin, au chapitre 7 est explicitée la technique dite du RWHT+P, tirant parti d'une forme nouvelle de la transformée réversible 2D Walsh-Hadamard.

Nous abordons par la suite une autre partie de la thèse consacrée à l'élaboration d'applications s'appuyant sur la méthode LAR. D'une part, une technique de codage conjoint source-canal, conçue autour de l'utilisation des méthodes pyramidales du LAR et de la transformée Mojette, est réalisée à des fins de sécurisation du processus de transmission des images compressées (chapitre 8). D'autre part, après avoir donné les éléments de base de la méthodologie AAA, nous nous intéressons à sa mise en œuvre (chapitre 9) : nous évoquerons la problématique de prototypage rapide du LAR sur des architectures cibles multi-composants hétérogènes, et les résultats associés.

La dernière partie de cette thèse concerne l'extension du LAR à l'encodage de vidéos (chapitre 10). Un bref état-de-l'art fait état des algorithmes d'estimation de mouvement disponibles : nous mettons en particulier l'accent sur les techniques fondées sur l'exploitation d'une représentation en régions. Enfin, sont exposées les méthodes que nous envisageons d'implanter dans un futur proche.





## Première partie

# Compression avec pertes d'images couleur - Représentation en régions



# Chapitre 1

## La méthode LAR (Locally Adaptive Resolution) : principes généraux

Les bases de la méthode de compression des images nommée LAR (Locally Adaptive Resolution) ont été jetées par Olivier Déforges [Déf04]. La technique était à l'origine élaborée à des fins de transmission à très bas débit pour des images (en niveaux de gris) reconstruites de très bonne qualité visuelle. Après une première application réalisée en 1999 [DR99], la méthode s'est enrichie de nombreuses fonctionnalités, telle l'introduction des composantes couleur, la représentation en régions (chapitre 2), ou encore les décompositions pyramidales (chapitres 5, 6 et 7), rendant totalement réversibles les opérations de codage.

Si l'ensemble de cette thèse est dévolue au LAR, il est essentiel d'en comprendre les fondements. Ce chapitre se propose donc de décrire les principes généraux, et s'inspire très largement, pour ce faire, de notre travail commun publié en 2005 dans la revue Traitement du Signal [DBBR04].

Le codeur LAR s'appuie sur une représentation en deux couches (§1) : le codeur spatial et le codeur spectral. Ces deux entités sont abordées successivement dans les sections 2 et 3. Nous présentons enfin des résultats de compression en section 4.

### 1 Le codec LAR simple : principe du codage en deux couches

L'idée fondatrice de la méthode LAR exploite la constatation suivante : la résolution locale peut s'adapter en fonction de l'activité observée dans l'image. Ainsi, lorsque la luminance s'avère localement uniforme, la résolution peut être abaissée. A l'inverse, si l'activité reste localement élevée, alors la résolution doit être plus importante.

D'autre part, une image  $I$  peut se voir comme la superposition de deux composantes, telles que

$$I = \bar{I} + \underbrace{(I - \bar{I})}_E \quad (1.1)$$

où  $\bar{I}$  représente une information globale de l'image estimée sur un support donné (de type moyenne locale par exemple), et  $E$  la variation locale autour de celle-ci (texture locale).

La dynamique de  $E$  est ainsi dépendante de deux facteurs essentiels :

1. l'**activité locale** dans l'image,

## 2. la dimension du support de $\bar{I}$ .

Si l'on admet de plus qu'une image peut grossièrement être considérée comme formée de zones relativement homogènes et de contours, alors  $E$  aura une faible dynamique dans les zones uniformes par la seule adaptation du support. En revanche,  $E$  sera de forte dynamique sur les contours dès lors que le support de  $\bar{I}$  sera supérieur à un pixel.

Le principe général de la méthode LAR repose sur un **codeur à deux couches** : couche spatiale pour le codage de  $\bar{I}$ , et spectrale pour l'image d'erreur  $E$  (texture). Par construction, le codec offre donc au moins deux niveaux de progressivité. La figure 1.1 montre le schéma de principe global.

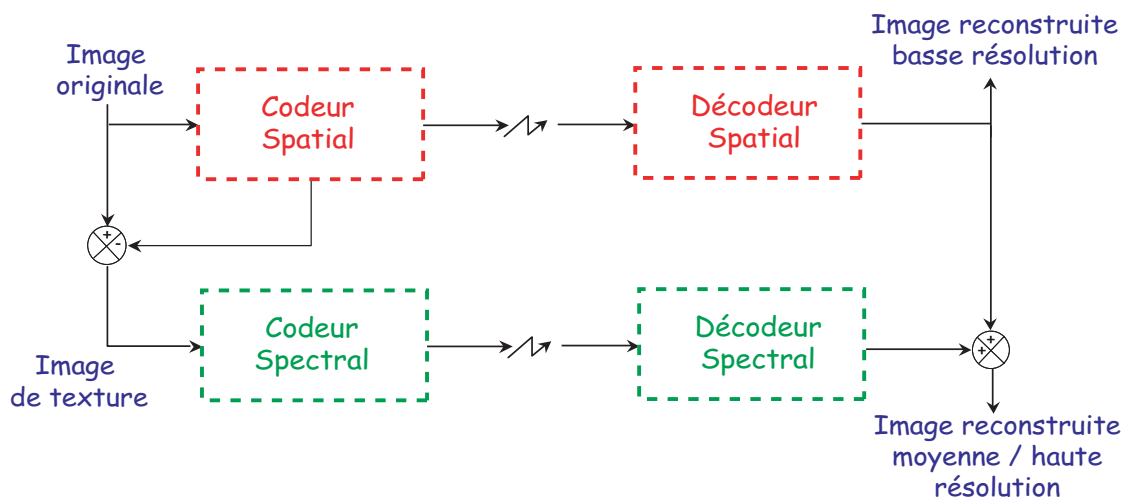


FIG. 1.1: Schéma global LAR à deux couches : codeurs spatial et spectral

Dans les sections suivantes, nous décrivons plus en avant le contenu des différentes couches de codage. L'espace de couleur choisi reste celui le plus classique en codage avec pertes, à savoir  $Y : Cr : Cb^1$ . Ce choix d'espace de représentation pour le codage des images couleur est motivé par la décorrélation de l'information contenue dans les composantes  $Y : Cr : Cb$  résultantes, par la repartition uniforme de l'entropie sur les composantes couleurs [DSE092], par la simplicité de la transformation, et enfin par la simplicité d'utilisation de cet espace de représentation (transformation linéaire, espace à valeurs entières).

## 2 Le codeur spatial

Le terme “spatial” indique que le procédé de représentation et de compression est réalisé directement dans le domaine spatial de l'image. En s'attachant à représenter et à compresser uniquement l'information globale dans l'image, ce codeur vise clairement les **forts taux de compression**. Son rôle est d'une part de distinguer les contours du reste de l'image, d'autre part

<sup>1</sup> $Y : Cr : Cb$  :  $Y$  exprime la luminance d'un pixel.  $Cr$  représente la différence entre la composante Rouge et la luminance, à savoir  $Cr = R - Y$ .  $Cb$  désigne la différence entre la composante Bleue et la luminance, soit  $Cb = B - Y$ . Les valeurs  $Y : Cr : Cb$  évoquent l'entrelacement de ces composantes

d'adapter le support de  $\bar{I}$  de telle sorte que l'image reconstruite soit visuellement acceptable avec une erreur  $E$  réduite dans les zones uniformes. La forme des supports considérés est ici celle de blocs carrés.

Le schéma bloc du codeur spatial est donné par la figure 1.2. Il s'appuie sur une **représentation de l'image avec un partitionnement à taille de blocs variable**, où chaque bloc prend pour valeur sa luminance moyenne. Si cette technique est *a priori* connue, nous montrons dans la suite les éléments distinctifs importants dans notre approche. L'ensemble de symboles générés (partitionnement, erreurs de prédiction, image d'erreurs...) par ces différentes phases de représentation et de codage de l'information est compressé par un codage arithmétique adaptatif propre.

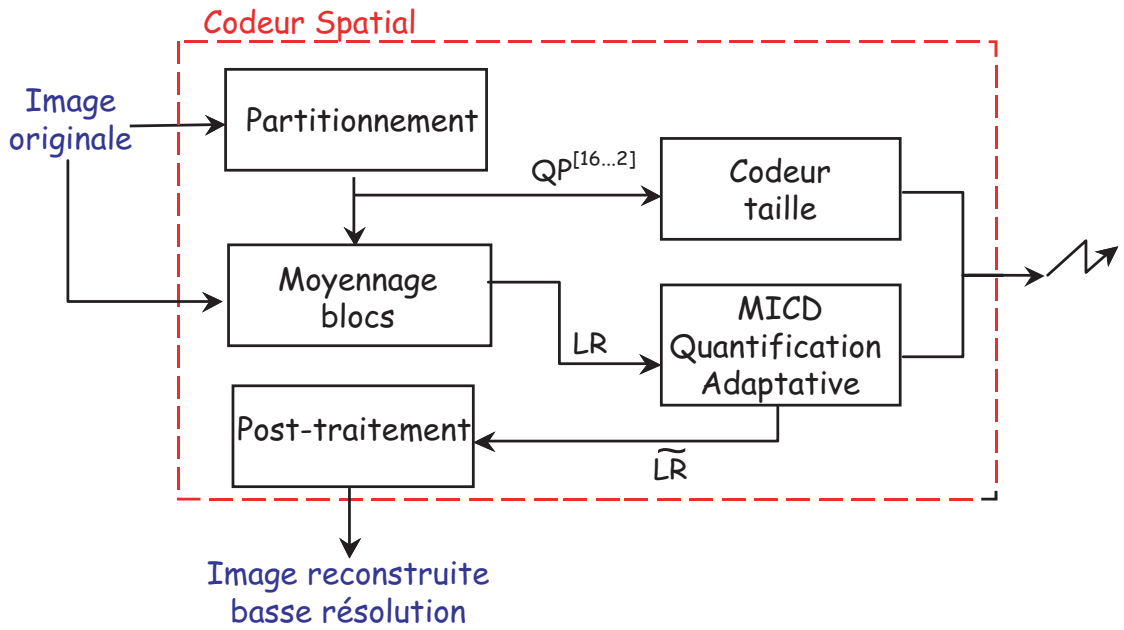


FIG. 1.2: Schéma de principe du codeur spatial

Les sections suivantes vont expliciter les différentes techniques mises en œuvre dans chacune de ces étapes.

## 2.1 Partitionnement

Tout système fondé sur une représentation à taille de blocs variable s'appuie sur un critère d'homogénéité et sur une topologie particulière de partition. Par la suite, on considère la partition **Quadtree**  $QP^{[N_{max}...N_{min}]}$ , où  $N_{max}$  et  $N_{min}$  représentent les tailles respectivement maximale et minimale autorisées des blocs.  $I(x, y)$  désigne un point de coordonnées  $(x, y)$  dans  $I$ , et  $I(b^N(i, j))$  le bloc  $b^N(i, j)$  dans l'image  $I$  tel que :

$$b^N(i, j) = \{(x, y) \in N_x \times N_y \mid N \times i \leq x < N \times (i + 1), \text{ et } N \times j \leq y < N \times (j + 1)\}. \quad (1.2)$$

Parmi les nombreux exemples existants, citons celui de H.264 en mode intra, autorisant une partition Quadtree  $QP^{[16...4]}$  (partitionnement de l'image en blocs de taille 4x4 à 16x16), où la

taille choisie est celle fournissant le meilleur rapport débit/distorsion d'un point de vue PSNR [H264 MPEG-4 10 AVC02]. Les méthodes fondées sur des structures en arbres fonctionnent depuis le plus haut niveau (ou taille maximale) par découpage des nœuds en fils, lorsque un seuil d'homogénéité est dépassé. Si plusieurs tests d'homogénéité ont pu être proposés dans la littérature [SS87, Str90], dans la majorité des cas ils s'appuient sur un calcul de distance de norme  $L_1$  ou  $L_2$ , entre la valeur d'un bloc et celle de ses quatre fils.

Nous proposons ici un critère différent fondé sur une détection de contours dont la justification sera donnée ultérieurement. Parmi les nombreux types de filtre possibles, nous avons opté pour un **gradient morphologique** (différence entre valeurs de luminance maximale et minimale sur un support donné), à la fois pour sa mise en œuvre rapide et récursive, et pour la propriété qui en découle de limitation de la valeur absolue de l'image de texture  $E$  (voir §2.2).

Considérons une partition Quadtree  $QP^{[N_{max} \dots N_{min}]}$ . Soient  $\min[I(b^N(i, j))]$  et  $\max[I(b^N(i, j))]$  respectivement les valeurs minimum et maximum du bloc  $I(b^N(i, j))$ .

L'image des tailles en tout point est donnée par :

$$Siz(x, y) = \begin{cases} N \in [N_{max} \dots N_{min}] & \text{si } |\max[I(b^N(\lfloor \frac{x}{N} \rfloor, \lfloor \frac{y}{N} \rfloor))] - \min[I(b^N(\lfloor \frac{x}{N} \rfloor, \lfloor \frac{y}{N} \rfloor))]| \leq Th \\ \text{et si } \exists(k, m) \in \{0, 1\}^2 / & |\max[I(b^N(\lfloor \frac{x+k}{N/2} \rfloor, \lfloor \frac{y+m}{N/2} \rfloor))] - \min[I(b^N(\lfloor \frac{x+k}{N/2} \rfloor, \lfloor \frac{y+m}{N/2} \rfloor))]| > Th \\ N_{min} & \text{sinon.} \end{cases} \quad (1.3)$$

où  $Th$  est le seuil d'homogénéité.

L'image des tailles donne alors directement une **carte de segmentation grossière** de l'image, où les blocs de taille  $N_{min}$  se retrouvent essentiellement sur les frontières et zones très texturées de l'image. Nous verrons dans la suite de cette étude que cette caractéristique est le fondement des différentes phases du procédé de codage (chapitre 2).

Dans le cas des **images couleur**, la solution adoptée consiste à définir une seule partition régulière pour l'ensemble des trois composantes Y :Cr :Cb pilotée par la taille minimale. L'image des tailles en tout point  $p(x, y) \in I$  s'obtient alors par

$$Siz(x, y) = \min[Siz_Y(x, y), Siz_{Cr}(x, y), Siz_{Cb}(x, y)]. \quad (1.4)$$

Les seuils  $Th$  pour la composante de luminance et celles de chrominance peuvent être fixés indépendamment dans notre schéma. Pour un même seuil  $Th$ , le minimum est majoritairement fourni par la composante Y. Dans la suite de cet article, nous nous sommes placés dans cette configuration.

## 2.2 Estimation des valeurs moyennes des blocs

Une image couleur basse résolution ( $LR_Y : LR_{Cr} : LR_{Cb}$ ) sera obtenue par le codeur spatial en représentant, pour chaque composante, chaque bloc par sa moyenne. En chaque point  $p(x, y)$ , l'image  $LR$  de chaque composante est définie par :

$$LR(x, y) = \frac{l}{N^2} \sum_{k=0}^{N-1} \sum_{m=0}^{N-1} I(\lfloor \frac{x}{N} \rfloor \times N + k, \lfloor \frac{y}{N} \rfloor \times N + m), \quad (1.5)$$

avec  $N = Siz(x, y)$ .

Comme la valeur moyenne du bloc est naturellement comprise entre les valeurs minimales et maximales, une propriété intrinsèque de la décomposition est que l'erreur de représentation  $E(x, y)$  se trouve bornée pour les blocs de taille supérieure à  $N_{min}$  :

$$E(x, y) = |I(x, y) - LR(x, y)| \leq Th, \quad (1.6)$$

pour tout  $p(x, y) \in P^{[N_{max} \dots N_{min}]}$ .

Pour chaque composante de l'image, l'entropie de l'erreur, l'erreur quadratique moyenne, ainsi que le PSNR, admettent ainsi une limite donnée par les relations suivantes :

$$\begin{aligned} H(E) &\leq \log_2(Th) \text{ bits}, \\ MSE &\leq Th^2, \\ PSNR &\geq 10 \log \frac{255^2}{Th^2} \text{ dB}. \end{aligned} \quad (1.7)$$

### 2.3 Encodage des moyennes par prédiction de type MICD

En plus du facteur de compression apporté par le sous-échantillonnage de l'image, le coût global du codage est réduit à travers une quantification des valeurs des blocs, et une phase prédictive de ces valeurs.

#### 2.3.1 Quantification des moyennes des blocs.

Les techniques de compression fondées sur l'optimisation débit/distorsion tentent de trouver le meilleur compromis entre coût de codage et erreur globale dans l'image d'un point de vue  $PSNR$  ou  $MSE$ , sans tenir compte de la perception visuelle humaine. Or des expérimentations ont démontré que l'œil s'avère beaucoup moins sensible à des variations de luminance et de chrominances dans des zones de type frontière (hautes fréquences visuelles [LM94, VWS90]) que dans des zones uniformes (basses fréquences visuelles). La loi de Ricco indique également que le seuil de détection d'un saut de luminance d'une zone est inversement proportionnel à la dimension de cette zone. Autrement dit, les dégradations visuelles engendrées sur un bloc par une quantification linéaire [BSB97], sont inversement proportionnelles à la taille de celui-ci.

Ce principe va simplement s'appliquer dans notre schéma de codage en opérant une quantification adaptée à la taille des blocs. Si  $q_N$  représente le pas de quantification pour les blocs de taille  $N \times N$ , une relation du type

$$q_N = \frac{q_{N/2}}{2} \quad (1.8)$$

entre les pas des blocs de taille  $N$  et  $N/2$  aboutit à une qualité visuelle quasiment constante sur toute l'image.



### 2.3.2 Prédiction des valeurs moyennes des luminances.

Le codage des luminances moyennes des blocs est ici directement réalisé dans le domaine spatial, par une approche de type MICD (Modulation d'Impulsions à Code Différentiel). Ce choix a été essentiellement motivé, d'une part par la simplicité du codage qui ne nécessite qu'un balayage régulier de l'image, d'autre part parce que la représentation en blocs obtenue fournit un *a priori* sur les zones d'activité, pouvant être utilisé pour adapter la prédiction.

Nous nous sommes inspirés des techniques existantes dans le domaine de la compression sans perte, où beaucoup de méthodes sont fondées sur ce type de prédicteur, afin de trouver le meilleur rapport efficacité/simplicité. Nous avons notamment implanté les prédicteurs MED (Median Edge Detector) de LOCO-I [WSS96], le prédicteur DARC (Differential Adaptive Run Coding) proposé dans [G<sup>+</sup>95]. Ce meilleur compromis a finalement été obtenu grâce à un simple prédicteur de Graham [Gra58] appliqué à notre contexte. Cette adaptation consiste dans le principe à réaliser une prédiction linéaire dans les zones homogènes, et non linéaire sur les contours. La prédiction est pilotée par le gradient local et permet d'optimiser le prédicteur en fonction du contexte. À tout point sommet  $p(x, y)$  du bloc  $b^N(x, y)$ , on définit la valeur estimée du bloc  $L\check{R}_Y(x, y)$  à partir des valeurs reconstruites  $L\check{R}_Y(x - k, y - m)$ ,  $(k, m) \in \{0, 1\}$ .

Le prédicteur est donné par la relation

$$L\check{R}_Y(x, y) = \begin{cases} L\check{R}_Y(x - 1, y) & \text{si } |L\check{R}_Y(x - 1, y - 1) - L\check{R}_Y(x, y - 1)| \\ & < |L\check{R}_Y(x - 1, y - 1) - L\check{R}_Y(x - 1, y)| \\ & \text{et si } A_N < |L\check{R}_Y(x - 1, y - 1) - L\check{R}_Y(x - 1, y)| \\ L\check{R}_Y(x, y - 1) & \text{si } |L\check{R}_Y(x - 1, y - 1) - L\check{R}_Y(x - 1, y)| \\ & < |L\check{R}_Y(x - 1, y - 1) - L\check{R}_Y(x, y - 1)| \\ & \text{et si } A_N < |L\check{R}_Y(x - 1, y - 1) - L\check{R}_Y(x, y - 1)| \\ \frac{(L\check{R}_Y(x - 1, y) + L\check{R}_Y(x, y - 1))}{2} & \text{sinon.} \end{cases} \quad (1.9)$$

$A_N$  est un paramètre croissant de  $N$ , avec  $A_1 = 0$ ,  $A_2 = 10$ ,  $A_4 = 20$ ,  $A_8 = 40$  et  $A_{16} = 80$ .

Les quantifications des erreurs de prédictions suivent le principe énoncé à la section précédente, avec une quantification adaptée à la taille du bloc. En appelant  $E_{LR_Y}(x, y)$  l'erreur de prédiction,  $\hat{E}_{LR_Y}(x, y)$  et  $\tilde{E}_{LR_Y}(x, y)$  respectivement les erreurs quantifiée et déquantifiée,  $q_N$  le pas de quantification appliqué aux blocs de taille  $N$ , les relations suivantes sont définies :

$$\begin{cases} E_{LR_Y}(x, y) = LR_Y(x, y) - L\check{R}_Y(x, y), \\ \hat{E}_{LR_Y}(x, y) = Q(E_{LR_Y}(x, y)) = \text{round} \left[ \frac{E_{LR_Y}(x, y)}{q_N} \right], \\ \tilde{E}_{LR_Y}(x, y) = Q^{-1}(\hat{E}_{LR_Y}(x, y)) = q_N \cdot \hat{E}_{LR_Y}(x, y), \\ L\check{R}_Y(x, y) = L\check{R}_Y(x, y) + \tilde{E}_{LR_Y}(x, y). \end{cases} \quad (1.10)$$

Les pas de quantification  $q_N$  fournis dans le tableau 1.1 correspondent aux valeurs possibles utilisables sans trop de dégradations de l'image. Nous avons empiriquement observé le fait suivant : les valeurs retenues sont telles qu'en les divisant par deux, la distorsion due à leur quantification devient imperceptible pour un affichage écran.

Taille	$q_N$
$16 \times 16$	2
$8 \times 8$	4
$4 \times 4$	8
$2 \times 2$	16
$1 \times 1$	32

TAB. 1.1: Valeurs de quantification selon la taille

### 2.3.3 Prédiction des valeurs moyennes des chrominances.

Un grand avantage du codage opérant directement dans le domaine spatial est de pouvoir se servir de la forte corrélation de structure existant entre les trois composantes. La composante  $LR_Y$  étant transmise en premier, le codage des valeurs moyennes des blocs de chrominance tire profit de cette information pour optimiser sa prédiction. L'estimation des valeurs prédites des chrominances en fonction des valeurs reconstruites des luminances est formalisée ci-dessous. Soit

$$GradMin_Y(x, y) = \min \left[ \left| \tilde{LR}_Y(x, y) - \tilde{LR}_Y(x, y-1) \right|, \left| \tilde{LR}_Y(x, y) - \tilde{LR}_Y(x-1, y) \right|, \left| \tilde{LR}_Y(x, y) - \left( \tilde{LR}_Y(x, y-1) + \tilde{LR}_Y(x-1, y) \right) / 2 \right| \right],$$

alors

$$\tilde{LR}_{Cr/b}(x, y) = \begin{cases} \tilde{LR}_{Cr/b}(x-1, y) & \text{si } |\tilde{LR}_Y(x, y) - \tilde{LR}_Y(x-1, y)| = GradMin_Y(x, y) \\ \tilde{LR}_{Cr/b}(x, y-1) & \text{si } |\tilde{LR}_Y(x, y) - \tilde{LR}_Y(x, y-1)| = GradMin_Y(x, y) \\ \frac{\tilde{LR}_{Cr/b}(x-1, y) + \tilde{LR}_{Cr/b}(x, y-1)}{2} & \text{sinon.} \end{cases} \quad (1.11)$$

Cette optimisation de la prédiction apporte un **gain très significatif de l'ordre de 20%** par rapport à un codage direct.

## 2.4 Post-traitement

Par construction, une image  $LR$  obtenue présente des "effets de blocs", surtout perceptibles sur la composante de luminance. Ceux-ci ne sont toutefois pas comparables aux artefacts des techniques fondées sur la décomposition de l'image en blocs de taille fixe comme JPEG, MPEG-2 ou MPEG-4. Ils apparaissent en effet ici essentiellement dans les zones uniformes représentées par valeurs de plateau, et sur les contours par un manque de résolution si  $N_{min} > 1$ . La possibilité d'un post-traitement peut s'envisager avec comme but de **lisser les zones homogènes tout en conservant les contours**. Une méthode d'interpolation linéaire adaptée au partitionnement de l'image a été élaborée. Les images obtenues sont alors de très bonne qualité visuelle sur les zones uniformes (blocs de taille  $8 \times 8$  et  $16 \times 16$ ) ainsi que sur les zones de contours (blocs de taille  $2 \times 2$  et  $4 \times 4$ ). Cependant, de meilleurs rendus visuels ont été obtenus par l'utilisation de l'algorithme d'interpolation directionnelle décrit par D. Muresan [MP01], fondé sur la théorie de la récupération optimale adaptative des valeurs manquantes. Cette technique sera décrite plus en détails au chapitre 5. Sur l'image de luminance 1.3, nous constatons le fait suivant : le post-traitement permet d'obtenir un lissage des zones homogènes tout en préservant les contours,

par l'application de l'interpolation le long de ceux-ci [Mur02].



a. Image source Lena  $512 \times 512$  - 8 bpp



b. Partition Quadtree : 13888 blocs



c. Image reconstruite basse résolution : 0.2 bits/pixel. Taux de compression : 40, PSNR 29.2 dB



d. Image reconstruite après post-traitement sur  $P^{[16...2]}$ , PSNR 29.7 db

FIG. 1.3: Résultats pour une partition  $P^{[16...2]}$ ,  $Th = 30$

## 2.5 Caractéristiques du codeur spatial

La première couche du codeur LAR, à savoir le codeur spatial, possède les qualités suivantes.

1. La méthode s'avère très **simple et rapide** à mettre en œuvre : l'essentiel de la complexité est concentré dans les phases de codage arithmétique et de post-traitement. Seule la phase d'estimation des tailles s'exécute au niveau pixel, les étapes de prédiction restant au niveau bloc.
2. La technique permet d'atteindre de **forts taux de compression** grâce à des quantifications simplement adaptées au système visuel humain.
3. La méthode à bas débit **simplifie** l'image en lissant les zones homogènes, tout en préservant correctement les contours.
4. Les approches classiques basées sur des transformations dans des espaces fréquentiels (DCT, ondelettes) rencontrent le phénomène suivant : les fortes quantifications appliquées, pour du

très bas débit, créent des distorsions fortes sur l'image reconstruite. Le traitement ici proposé s'effectue directement dans l'espace spatial. Ainsi, **les distorsions observées sont très atténuées**, en comparaison avec les techniques précédemment évoquées.

5. Lorsque  $N_{min} > 0$ , un sous-échantillonnage d'un facteur  $N_{min}$  est naturellement possible. L'image sera alors reconstruite à la résolution maximale dans laquelle les contours se représentent exactement.

Ce codeur peut s'utiliser seul pour de forts taux de compression compris entre 80 et 30, soit  $Th \in [20 \dots 60]$ . Lorsque l'on désire augmenter la qualité, la technique atteint ses limites. En effet, l'image est alors représentée par des blocs plus petits, même dans les zones relativement uniformes. Le bruit de quantification devient par la suite beaucoup plus perceptible. Une manière plus efficace, apportée par le codeur spectral présenté ci-après, va consister à compresser l'image d'erreur ou de texture, par une technique opérant dans des espaces transformés.

### 3 Le codeur spectral

L'image d'erreur  $E$  issue de la représentation du codeur spatial est compressée dans un espace transformé fréquentiel, par une seconde couche, appelée codeur spectral. Le support de  $E$  considéré est le même que celui de  $\bar{I}$ , ce qui permet une caractérisation *a priori* de  $E$ , et une adaptation du schéma de compression.

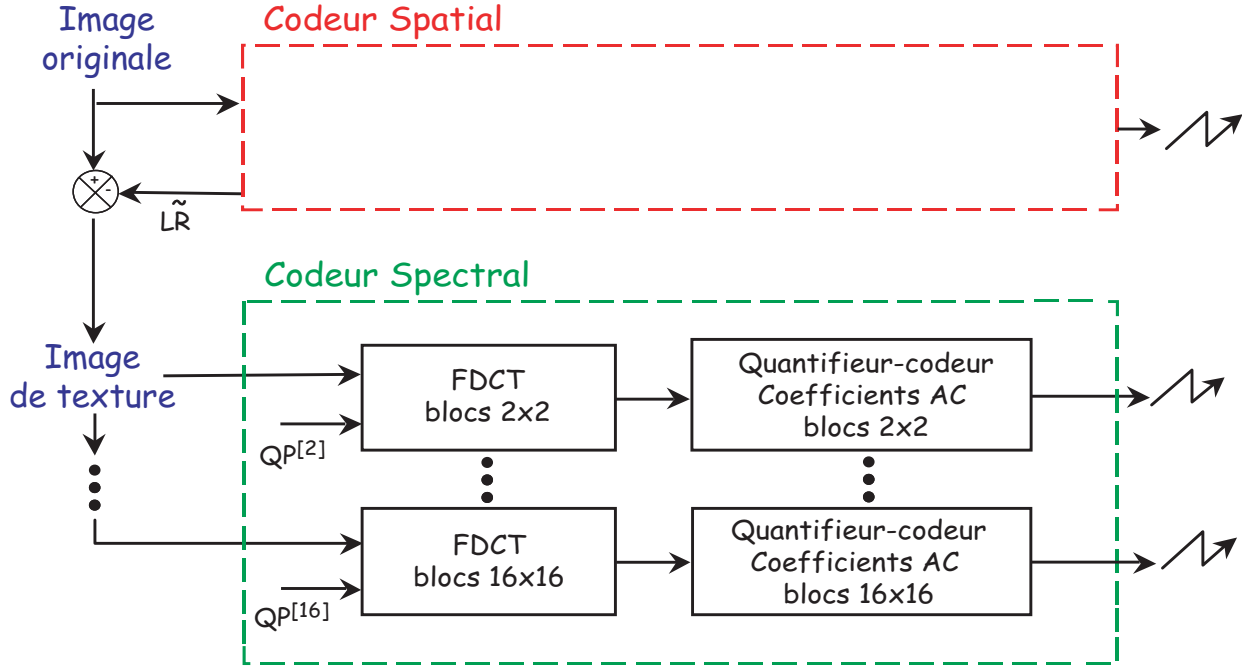
Dans la suite de ce document, nous appellerons **contours** les blocs situés dans la partition  $QP^{[N_{min}]}$ , et par complémentarité **texture locale** les blocs contenus dans  $QP^{[N_{max} \dots N_{min}]}$ . Notons que si  $N_{min} = 1$ , alors le codage des contours s'effectue dans le codeur spatial.

La technique de codage définie se base sur une **approche DCT à taille de blocs variable**, où la taille est établie par la partition  $QP^{[N_{max} \dots N_{min}]}$  de l'étage précédent (voir fig. 1.4). Seuls les coefficients  $AC$  nécessitent une transmission puisque le premier codeur fournit déjà la luminance moyenne (coefficient  $DC$ ).

Les étapes majeures du procédé sont :

- l'application d'une transformation DCT de taille adaptée au bloc,
- le codage des coefficients : un balayage zig-zag intra-bloc, suivi d'un codage des valeurs non nulles par "longueur de plage" (LDP), incluant les indicateurs spécifiques de fin de bloc et de "plage maximale",
- la quantification : la table de quantification s'adapte à chaque taille de bloc.

Par construction, le schéma de codage est progressif, autorisant des transmissions séparées des flux en fonction des tailles des blocs. Ainsi, est-il possible par exemple de n'améliorer que les contours par l'envoi de la texture des blocs de taille  $N_{min}$ . La figure 1.5 illustre le processus, où une transmission suivant l'ordre croissant des tailles conduit en premier à améliorer les frontières, puis en final à restaurer le fond. Le procédé inverse peut bien entendu être implanté.

FIG. 1.4: Schéma de principe du codeur spectral pour une partition  $QP[16...2]$ 

### 3.1 Énergie des blocs à taille fixe / variable

Par construction, les blocs contenant la texture locale présentent une erreur bornée, l'essentiel des erreurs se concentrant dans les blocs de petite taille. Par voie de conséquence, l'énergie moyenne des coefficients *AC*, pour la partition  $QP[N_{max}...N_{min}]$ , reste très inférieure à celle trouvée dans des approches classiques à taille de blocs fixe (voir figure 1.6). De plus, les coefficients se regroupent très majoritairement près de la composante continue. Après quantification, les coefficients à coder sont à la fois moins nombreux et de dynamique moindre.

### 3.2 Quantification des coefficients *AC*

Nous avons opté dans un premier temps pour les tables de quantification issues de JPEG, en les tronquant ou extrapolant suivant la dimension des blocs. Cette solution a montré ses limites, particulièrement pour des compressions à hauts débits. La raison réside dans le fait que les caractéristiques de quantification pour du JPEG ont été établies pour prendre en compte tout aussi bien des blocs à forte ou à faible activité. Par construction, l'énergie des blocs traités ici est limitée, ce qui remet en cause les hypothèses sur lesquelles s'appuie JPEG.

Nous donc opté pour un autre schéma très simple de quantification linéaire à base de deux paramètres  $Q_N$  et  $\Delta_{Q_N}$  tels que pour un coefficient *AC* sur la  $k^{eme}$  diagonale d'un bloc  $N \times N$ , son pas de quantification soit donné par :

$$Q = Q_N + k \cdot \Delta_{Q_N} \quad (1.12)$$



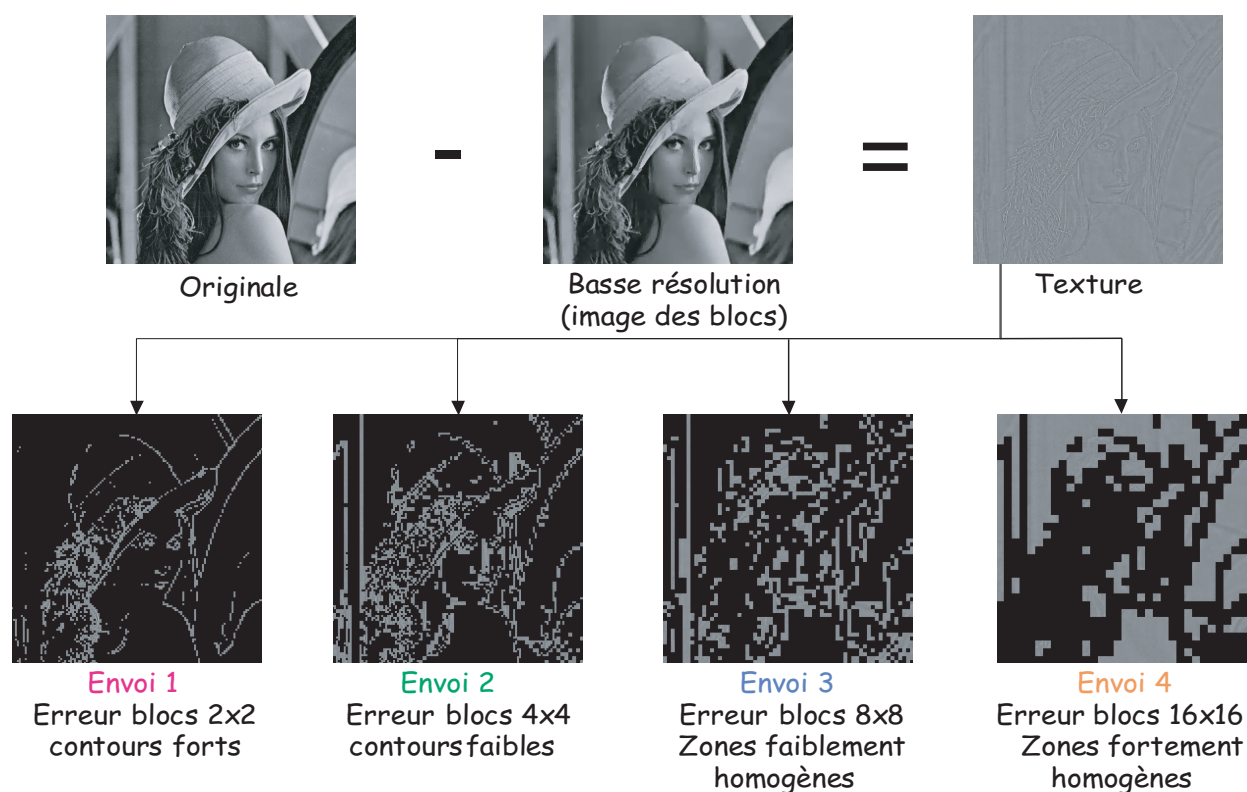


FIG. 1.5: Principe de progressivité sémantique

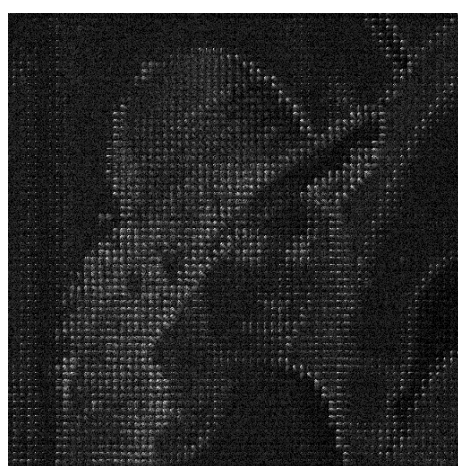
a. JPEG : partition  $P^{[8]}$ b. LAR : partition  $P^{[16...2]}$ 

FIG. 1.6: Répartition de l'énergie : comparaison entre partitions à taille fixe et taille variable

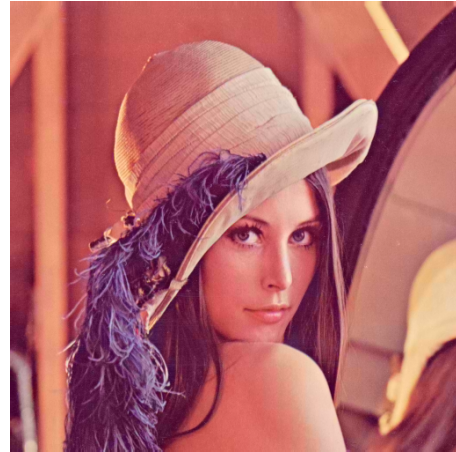
## 4 Résultats codec LAR simple

Dans une optique de codage d'image couleur, se révèle le fait que le codeur spatial seul est à même de coder et représenter les images de chrominances avec une bonne qualité. En particulier, dans les zones uniformes (blocs de taille supérieure à  $N_{min}$ ), le sous-échantillonnage plus important réalisé n'est pratiquement pas perceptible. Par ailleurs, pour une partition dans laquelle  $N_{min} = 2$ , coder les chrominances par le seul codeur spatial revient "au mieux" à une compression dans le format 4:2:0, où  $Cr$  et  $Cb$  sont sous-échantillonnées d'un facteur 2 dans les deux directions. Enfin d'un point de vue complexité, l'intérêt principal est le suivant : l'extension du schéma de codage à trois composantes ne multiplie pas par trois la complexité du système.

Pour illustrer cette bonne qualité de codage des chrominances, la figure 1.7 présente des images où seules les deux composantes chromatiques ont été compressées en utilisant uniquement le codeur spatial, l'image reconstruite incluant ici la composante  $Y$  d'origine.



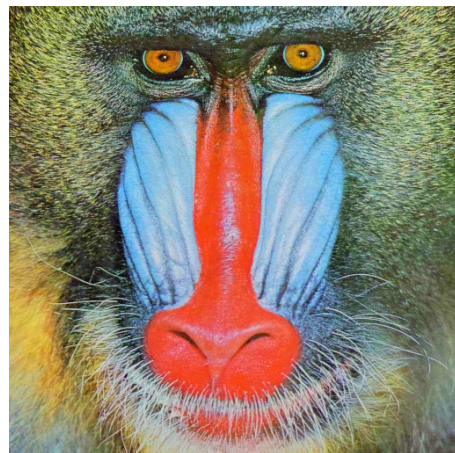
a. Image Lena, Cr/Cb 16 bpp



b. Cr+Cb codées blocs : 0.063 bpp



c. Image baboon, Cr/Cb 16 bpp



b. Cr+Cb codées blocs : 0.226 bpp

FIG. 1.7: Exemples d'images reconstruites avec codage des chrominances par le codeur spatial ( $Y$  source non modifiée)

De fait, le critère de débit/distorsion n'apparaît pas réellement comme étant un indicateur rendant compte de la qualité des images reconstruites. Pour leur évaluation subjective, des images

issues de notre schéma de compression ont été intégrées à une série de tests comparatifs pour juger de la qualité d’images compressées par différents schémas de codage. Ces tests ont été réalisés par P. Le Callet de l’IRCCyN : l’objectif était d’en extraire des lois pour un système automatique de mesures de qualité.

Cette étude a porté sur trois méthodes JPEG, JPEG2000 (codec *ImagXpress*, simple profile) et le LAR, et a été effectuée sur huit images standard (*Lena*, *Baboon*, *Boats*, *House*, *Pepper*, *Fruits*, *Airplane*, *Barbara*) compressées à différents débits. L’environnement était parfaitement standardisé en termes de distance à l’écran, de luminosité, de calibrage du moniteur, d’éclairage ambiant et de température de couleur pour une évaluation rigoureuse.

Le protocole élémentaire d’évaluation d’une image se déroulait ainsi :

1. image originale durant six secondes,
2. gris uniforme durant deux secondes,
3. image à évaluer pendant six secondes,
4. gris uniforme durant deux secondes.

Chaque observateur (quatorze au total) devait ensuite donner une note située sur une échelle de qualité entre un (très mauvaise qualité) et cinq (très bonne qualité). Sur huit séries d’images différentes, le LAR s’est montré supérieur pour sept d’entre elles. Seule l’image baboon très texturée a fait exception et cela pour les bas débits. Nous montrons (figure 1.8) les résultats obtenus pour trois de ces séries.

Notons toutefois que les trois codeurs n’ont pas les mêmes caractéristiques d’un point de vue progressivité : le mode de codage JPEG choisi est celui du non progressif. Les résultats débits/qualité correspondants ont été obtenus en faisant varier le paramètre de quantification, pour des codages distincts. A l’opposé, le codeur JPEG2000 est entièrement progressif et les courbes associées reflètent l’évolution continue de la qualité possible. Le codage LAR se situe ici entre les deux, avec une progressivité du codage par paliers. Par ailleurs, les codeurs JPEG et LAR admettent une limite de compression à bas débits qui, pour le LAR, est fixée par la première étape de codage spatial. Cette limite se révèle d’autant plus basse que l’image est faiblement texturée. La figure 1.9 illustre à titre d’exemple, la qualité visuelle de notre schéma de codage.

## 5 Conclusion

Le codec LAR présenté ici constitue la version la moins élaborée et donc bien évidemment la moins complexe. Une alternative, à la fois au codage MICD de la couche spatiale et au codeur spectral dans son ensemble, sera explicitée au cours des chapitres 5, 6 et 7. Il s’agira d’une décomposition pyramidale originale avec et sans perte, dite à “contexte enrichi”, élargissant la capacité d’encodage progressif au codeur spatial. Cette forme de codage permettra en grande partie de lever les limitations du LAR dans sa forme simple, à savoir une progressivité par paliers beaucoup plus fine, et un codage possible à très bas débits par diminution de la résolution spatiale.

Les principes de base resteront toutefois les mêmes (partition suivant l’activité), et la technique de représentation en régions présentée dans le chapitre suivant est applicable dans les



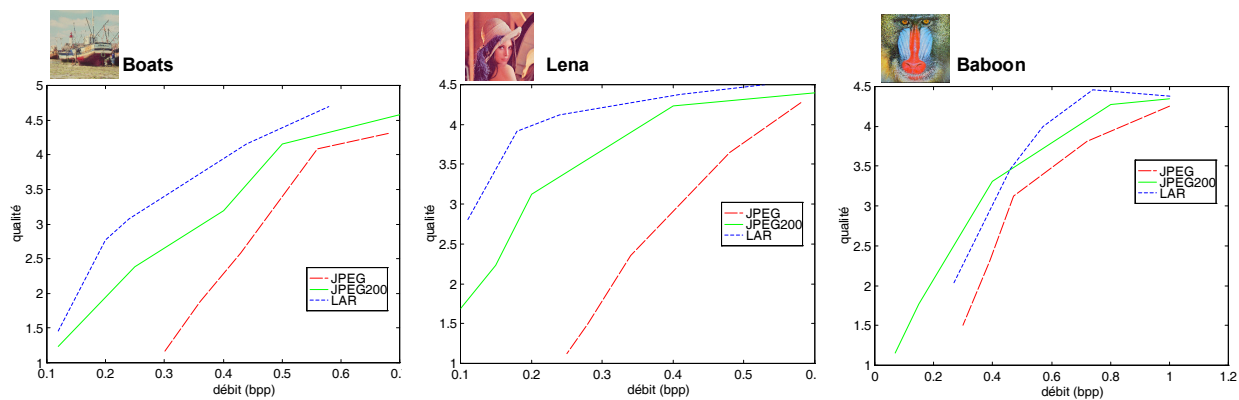


FIG. 1.8: Résultats tests comparatifs de perception visuelle

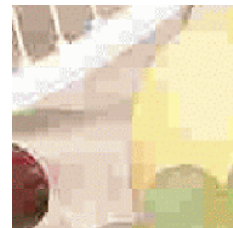
deux versions du codec (LAR simple ou pyramidal).



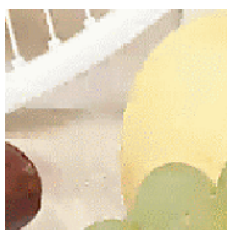
*a. Image source Bike 24 bpp*



*b. Image codée JPEG : 0.25 bpp*



*c. Image codée LAR : 0.25 bpp*



*d. Image codée JPEG2000 : 0.25 bpp*



*e. Image codée LAR : 0.50 bpp*



*f. Image codée JPEG2000 : 0.50 bpp*



FIG. 1.9: Résultats comparatifs de qualité visuelle



## Chapitre 2

# Représentation en régions auto-extractibles

Dans une images, les régions sont définies comme des **parties convexes partageant une propriété commune**. Les objets peuvent quant à eux être vus comme des ensembles de régions qui représentent des entités sémantiquement cohérentes d'une image [EK98]. Pour des représentations en régions, deux types d'information sont nécessaires : **la forme** (les contours) et **le contenu** (la texture). Un troisième élément peut être ajouté dans le cas de séquences vidéo, à savoir **le mouvement**. Le principe d'une représentation en objets fait partie intégrante du standard MPEG-4 [MPEG-4 Committee00] pour la compression et la manipulation de flux vidéo, et est un pré-requis nécessaire dans MPEG-7 pour l'indexation [Gro99].

Le chapitre précédent fait état d'une méthode de compression des images efficace, dénommée LAR. Le codage s'appuie sur une représentation de l'image exploitant un partitionnement à taille de blocs variable. Cette information, constituant une grille définie par observation de l'activité locale, peut également se concevoir comme une première carte de segmentation grossière de l'image, où chaque région (chaque bloc) se révèle cohérente du point de vue de sa forme et son contenu.

L'objectif des travaux présentés ici est de réaliser, partant de la constatation précédente, une représentation en régions homogènes en termes à la fois de forme et de contenu. Notre approche repose en grande partie sur l'exploitation de l'information couleur pour l'obtention d'une représentation en régions de bonne qualité, tout en exploitant cette même représentation en régions pour un codage très bas débit des images de chrominance. Une grande originalité de ce schéma de codage est de ne pas recourir à l'envoi d'une carte de segmentation pour la représentation en régions, mais de la déduire à partir d'une image de luminance codée à bas débit. Ce procédé est appelé “**Représentation en Régions Auto-Extractibles**”. Ce sujet a fait l'objet d'une publication en revue [DBBR04]. La notion de Région d'Intérêt, évoquée dans cette étude, a elle aussi été matière à publication [DBR03].

Le chapitre s'organise comme suit. La première section s'interroge sur la légitimité d'une représentation en régions à des fins de codage avec pertes. Nous présentons également le principe de notre codeur LAR associé à une représentation en régions efficace. Nous exposons en section 2 les bases théoriques des méthodes de segmentation par graphes d'adjacence, avant de donner l'algorithme propre au codeur LAR (§3, description à coût nul). Les applications directes de cette segmentation sont introduites au paragraphe 4. Enfin, nous apportons une amélioration

importante au codeur par la définition d'une méthode de segmentation intégrant un contrôle des opérations de fusion au moyen des composantes de chrominance (§5).

## 1 Préambule

### 1.1 Une représentation en régions : pourquoi ?

La première génération de codeurs d'images et de vidéos s'appuie essentiellement sur la théorie de l'information. La principale limitation de ses performances en termes de compression est due au caractère non-stationnaire des images naturelles, aboutissant à une distribution variable de l'information dans l'espace et dans le temps. De plus, les structures de représentation utilisées, indépendantes du contenu, comme la grille cartésienne conventionnelle ou les pavages en blocs carrés réguliers (JPEG ou MPEG), ne permettent pas de prendre en compte ces non-stationnarités, et ne peuvent par conséquent servir comme structure de données efficace pour les images et la vidéo. En introduisant une modularité dans la partition de l'image, le nouveau standard H.264 ou MPEG4-AVC démontre que les performances de codage peuvent alors être fortement accrues [H264 MPEG-4 10 AVC02].

**Des améliorations ont pu être apportées par rapport aux schémas classiques, grâce à la représentation des données visuelles sous forme de régions, définies par leurs contours et leurs textures, et correspondant potentiellement à des objets ou parties d'objets dans l'image.** Les méthodes de codage s'appuyant sur ce concept sont couramment appelées de seconde génération [KIK85, TK96].

Une approche région tend par ailleurs à combler le fossé séparant les systèmes numériques et le système visuel humain (SVH) pour le traitement d'une image et sa **perception**. Une telle approche fournit également de nouvelles fonctionnalités telles que l'interaction entre objets ou régions, la composition de scènes, etc...

Un autre atout important réside dans la possibilité pour un schéma de compression de favoriser des zones visuellement sensibles de l'image tout en négligeant les parties moins significatives. Ce concept porte le nom de compression par **Région D'Intérêt** ou ROI (Region Of Interest). Une telle approche s'adapte bien au développement de nouvelles applications à travers les réseaux (consultation de grandes bases de données d'images, vidéosurveillance et vidéoconférence) [PZ95]. Les bandes passantes limitées des canaux actuels comparées aux volumes de données nécessaires pour la représentation de ces images amènent à établir un compromis entre le débit et la distorsion (qualité) de l'image. Lorsqu'une ROI est identifiée et définie, ce compromis peut être réalisé localement de telle sorte que la compression à l'intérieur de la ROI n'introduise que peu de distorsion pour une bonne qualité visuelle, alors que le reste de l'image peut être représenté avec une moindre qualité.

Malgré tous les attraits que présentent les approches régions, il faut reconnaître que les standards actuels s'appuient toujours sur des techniques traditionnelles de transformation de l'information. Les raisons se déclinent en quatre points.

1. La description de la forme (le plus souvent un polygone) constitue un surcoût qui peut rapidement devenir significatif à très bas débit. Pour limiter ce coût, il est donc aussi nécessaire

de limiter le nombre de régions, et par conséquent de se contenter d'une description rudimentaire de la scène.

2. Les méthodes basées régions s'attachent en général essentiellement à la composante "forme", et négligent le plus souvent la composante "contenu". Dès lors, cette forme est indépendante de la représentation utilisée pour coder le contenu.
3. Une représentation en régions induit forcément une phase de segmentation de l'image. Or cette étape constitue en général un verrou important pour la réalisation de systèmes de traitement temps-réel.
4. Le schéma de codage classique impose que seul le codeur soit à même de définir la représentation en régions, le décodeur ne pouvant avoir un rôle de décideur. Cela exclut alors des applications du type consultation de base de données d'images avec définition, par l'utilisateur, d'une région d'intérêt au décodeur.

## 1.2 Principe du codeur LAR associé à la description en régions à coût nul

Pour obtenir une représentation en régions à coût de description nul, la seule solution réside dans une transmission d'une image compressée, suivie d'une segmentation identique au codeur et au décodeur. N'importe quel schéma de codage peut alors convenir lorsque l'image est codée à une bonne qualité (moyen/haut débit). Toutefois les méthodes standard engendrent à très bas débit des dégradations telles que le procédé de segmentation n'est plus envisageable. Pour sa part, notre approche, fondée sur une représentation de l'image cohérente en termes de contours et de zones uniformes, se prémunit de ces dégradations pénalisantes.

Ainsi, en rapprochant notre méthode à la technique de segmentation fondée Quadtree exposée dans [WRK91], nous avons développé **une technique de segmentation en régions s'appuyant sur la représentation du codeur spatial  $LR_Y$** . L'opération de découpage en blocs ("split") ayant déjà été effectuée à travers le codeur spatial, il nous suffit de réaliser l'opération de fusion ("merge") sur les blocs au codeur et au décodeur pour disposer d'une carte de segmentation identique. La reconstruction directe des régions à partir de la représentation en blocs apporte une compatibilité entre forme et contenu des régions, et exploitée par la suite pour du codage par ROI de la texture et le codage par régions de la chrominance. Notons qu'il n'est pas question ici de vouloir proposer la meilleure technique de segmentation qui soit, mais bien de développer une méthode globale de représentation à partir d'images compressées bas débit. Nous avons également cherché à modifier les approches existantes dans le but d'une simplification du procédé de segmentation pour, à terme, envisager une implantation temps réel de notre schéma de codage.

## 2 Méthodes de segmentation en régions par graphes d'adjacence

### 2.1 Problématique

Soient  $S = \{(x, y) | 1 \leq x \leq N_x, 1 \leq y \leq N_y\}$  les coordonnées spatiales des pixels dans une image de  $N_x$  lignes et  $N_y$  colonnes. La segmentation d'une image en  $K$  régions  $R_k^K$  consiste à



trouver la partition  $\Delta^K$  de  $S$  telle que :

$$S = \bigcup_{k=1}^K R_k^K, \quad (2.1)$$

avec  $R_i^K \cap R_j^K = \emptyset, \forall (i, j) \in \{1 \dots K\}^2$  pour  $i \neq j$ .

Notons  $S^K$  l'ensemble des régions dans la partition  $\Delta^K$ . Partant d'une partition initiale  $\Delta^{K_0}$  ( $K_0 \leq N_x \times N_y$ ), le but du procédé de segmentation est de transformer  $\Delta^{K_0}$  en une nouvelle partition  $\Delta^K$  ( $K < K_0$ ) selon un critère d'homogénéité, et à travers des séquences de fusions de régions. Dans notre cas,  $\Delta^{K_0}$  vaut  $P^{[N_{max} \dots N_{min}]}$ , et  $S^{K_0}$  correspond aux blocs de luminance issus du codeur spatial.

Partitionner l'ensemble des régions élémentaires  $S^{K_0}$  en sous-ensembles peut se traduire par la recherche d'une relation d'équivalence  $\mathfrak{R}$  sur  $S^{K_0}$ . Ces sous-ensembles formés constituent alors des classes d'équivalence. Dans la suite, nous noterons  $\left[R_i^{K_0}\right]_{\mathfrak{R}^K}$  une région de  $S^K$  dans la partition  $\Delta^K$ , initialement associée à la région  $R_i$  de  $S^{K_0}$ .

## 2.2 Graphe d'adjacence

De manière naturelle, les régions reconstruites doivent former des ensembles spatialement connexes. Aussi, la relation d'adjacence est-elle au cœur des principes de segmentation. Nous appellerons par la suite  $A_i^K$  l'ensemble des régions connexes à  $\left[R_i^{K_0}\right]_{\mathfrak{R}^K}$  dans la partition  $\Delta^K$ .

La structure de données classique pour représenter des partitions est le "Region Adjacency Graph" (RAG) [SHB93]. Le  $RAG^K$  d'une K-partition est défini comme un graphe non orienté,  $G^K = (V, E)$ , où  $V = \{1, \dots, K\}$  est l'ensemble des sommets et  $E \subset V \times V$  est l'ensemble des arêtes. Chaque région est représentée par un sommet du graphe, et entre deux sommets (régions)  $R_i^K, R_j^K \in V^2$  il existe une arête  $(i, j)$  si les régions sont adjacentes.

## 2.3 Classification hiérarchique et métrique

Fusionner les régions selon un critère d'homogénéité se ramène en général à un problème de classification hiérarchique, consistant à chercher les éléments les plus proches au sens d'une distance  $D$ , puis à mesurer les agrégations entre les classes suivant un critère  $Crit$  donné. La hiérarchie est dite *indicée* si pour toute partie  $H$  de la hiérarchie, la relation d'inclusion  $H \subset H'$  implique que  $D(H) \leq D(H')$ . Un niveau hiérarchique donné correspond alors à la fusion entre un sommet et un ensemble de sommets connexes.

Les méthodes de segmentation utilisant ce schéma fonctionnent en général sur un même modèle, où les fusions sont effectuées deux à deux suivant un critère de distance minimale [HEMK98, BG89]. Le critère de distance minimale consiste simplement à fusionner les deux régions au niveau courant de hiérarchie, dont la distance est minimale. La hiérarchie est par conséquent indicée pour tous les niveaux.

La distance retenue est en général fondée sur la différence des luminances moyennes des régions, mais certaines méthodes proposent des métriques moins rudimentaires comme par exemple des caractéristiques de texture [MAIS96].

L'intérêt d'une approche par fusion selon la distance minimale réside dans le fait que l'on puisse exactement contrôler le nombre final de régions. Par construction, les régions fusionnées sont également les plus proches du point de vue de la distance choisie. Cet avantage n'est toutefois que relatif, car le nombre nécessaire de régions pour décrire "correctement" une image reste bien sûr dépendant de la complexité de celle-ci.

Le point faible majeur de cette approche tient surtout dans sa complexité. Même si des algorithmes rapides, fondés sur les listes d'attente, existent pour classer les distances, la méthode n'en demeure pas moins très coûteuse en temps [HEMK98]. Ceci est d'autant plus vrai lorsque l'ultramétrie (nouvelle distance après fusion de deux régions) est fondée sur une nouvelle mesure de distance en fonction des régions fusionnées, nécessitant alors aussi un reclassement.

D'autre part, les méthodes classiques considèrent une symétrie des distances entre deux régions. Or pour éviter la sur-segmentation sur les contours tout en préservant l'intégrité des zones homogènes, un critère prenant en compte la taille des régions peut, pour une même valeur de distance, permettre de favoriser davantage la fusion de petites régions par rapport à des plus grandes.

### 3 Méthode de segmentation proposée : description à coût nul

Les principales modifications apportées ici aux schémas classiques de segmentation construits sur des RAG concernent d'une part la notion de distance pondérée, d'autre part la distance conjointe moyenne/gradient, et enfin le procédé de classification lui-même.

#### 3.1 Distance pondérée

Pour rendre le pouvoir de fusion dépendant de la taille des régions, nous avons introduit une pondération des distances en fonction de la surface des régions. Si  $Cost\left(\left[R_i^{K_0}\right]_{\mathbb{R}^K}, \left[R_j^{K_0}\right]_{\mathbb{R}^K}\right)$  définit la distance entre deux classes, alors la distance pondérée est donnée par :

$$Cost'\left(\left[R_i^{K_0}\right]_{\mathbb{R}^K}, \left[R_j^{K_0}\right]_{\mathbb{R}^K}\right) = Cost\left(\left[R_i^{K_0}\right]_{\mathbb{R}^K}, \left[R_j^{K_0}\right]_{\mathbb{R}^K}\right) \log_{10}\left(Surf\left(\left[R_i^{K_0}\right]_{\mathbb{R}^K}\right)\right) \quad (2.2)$$

où  $Surf\left(\left[R_i^{K_0}\right]_{\mathbb{R}^K}\right)$  désigne la surface de la région  $\left[R_i^{K_0}\right]_{\mathbb{R}^K}$ . Cela entraîne la relation suivante :

$$\begin{aligned} Surf\left(\left[R_i^{K_0}\right]_{\mathbb{R}^K}\right) > Surf\left(\left[R_j^{K_0}\right]_{\mathbb{R}^K}\right) &\Leftrightarrow \\ Cost'\left(\left[R_i^{K_0}\right]_{\mathbb{R}^K}, \left[R_j^{K_0}\right]_{\mathbb{R}^K}\right) > Cost'\left(\left[R_j^{K_0}\right]_{\mathbb{R}^K}, \left[R_i^{K_0}\right]_{\mathbb{R}^K}\right) \end{aligned} \quad (2.3)$$

Un effet immédiat de cette non symétrie des distances est que le *RAG* n'est plus un graphe non orienté : entre deux sommets connexes se trouvent deux arêtes, pondérées différemment suivant le sens de la relation d'adjacence.

#### 3.2 Distances pondérées par la moyenne et le gradient.

La distance couramment utilisée est celle de la différence des valeurs moyennes de niveau de gris (ou couleur) des régions. Elle permet de juger du degré de similarité des régions d'un



point de vue représentation par valeur plateau. Ce coût a également été retenu ici sous la notation  $Cost_M \left( \left[ R_i^{K_0} \right]_{\mathbb{R}^K}, \left[ R_j^{K_0} \right]_{\mathbb{R}^K} \right)$ . La gestion de l'ultramétrie au sein de la hiérarchie est particulièrement simple à implanter, puisqu'il suffit de conserver les deux caractéristiques suivantes : la surface et la valeur moyenne d'une région.

Il est bien connu qu'un critère de fusion des régions, fondé uniquement sur les valeurs moyennes, fonctionne mal dans des zones uniformes comportant un gradient local. La segmentation engendre alors des phénomènes dits de faux contours. Pour y remédier, nous avons donc ajouté une distance  $Cost_{Gr}$  à cette distance  $Cost_M$ .

Cette distance  $Cost_{Gr}$  est fondée sur une mesure de gradient au niveau des frontières entre les régions. Estimer le gradient implique de considérer la relation d'adjacence non pas uniquement au niveau courant de la partition, mais bien au niveau local de la partition initiale  $\Delta^{K_0}$ . Ceci est particulièrement simple pour une partition Quadtree où la longueur de frontière commune entre deux blocs correspond alors simplement à la taille minimale entre les deux blocs. La fonction de coût fondée gradient est évidemment plus complexe à mettre en œuvre que celle exploitant la moyenne. Malgré tout, cette topologie particulière permet d'envisager une solution algorithmique qui ne nécessite pas de traitement directement au niveau de l'image, mais uniquement des opérations dans la structure de représentation associée aux graphes d'adjacence.

Nous avons exprimé dans un premier temps la distance totale comme une somme pondérée des deux distances  $Cost_M$  et  $Cost_{Gr}$ , de manière à évaluer la qualité de la segmentation en fonction du poids de chacune. De cette étude est ressorti le fait que les meilleurs résultats s'obtenaient lorsque les contributions des deux distances étaient sensiblement égales. La distance totale a donc simplement été fixée comme la moyenne entre  $Cost_M$  et  $Cost_{Gr}$ .

### 3.3 Critère de fusion.

Nous avons déjà souligné que le critère de distance minimale couramment utilisé, présente une complexité importante. Plus encore, son utilisation conjointement à des distances asymétriques conduit dans notre contexte à s'intéresser initialement aux petits blocs, pour une construction des régions depuis les contours jusqu'aux zones homogènes : une telle démarche n'est pas cohérente. Nous avons donc mis en place une méthode plus rapide, et surtout mieux adaptée aux distances asymétriques. Pour chaque balayage du graphe, on détermine simplement pour chaque région celle la plus proche d'un point de vue distance, puis la fusion est effectuée si la distance est inférieure à un seuil. Le procédé est réitéré jusqu'à ce qu'aucune fusion ne soit obtenue. Nous donnons ici l'algorithme de manière schématique (algorithme 2.1). La variable  $Th_{Cost}$  désigne le paramètre de réglage du degré de simplification de l'image.

### 3.4 Hiérarchie indicée

La méthode proposée n'engendre pas de hiérarchie indicée, puisque les niveaux dans la hiérarchie vont dépendre essentiellement du nombre de fusions réalisées pour une région. Toutefois, en réitérant le procédé pour des valeurs croissantes de seuils de segmentation, une hiérarchie indicée s'obtient, avec autant de niveaux que de seuils. *La représentation en régions reconstruite*

**Algorithme 2.1** Algorithme de fusionAlgorithme de fusion  $K_0$  : partition initiale (blocs) $Nb_{fusions} = 0$ ; $K = K_0$ ;**Répéter** $Nb_{fusions\_prec} = Nb_{fusions}$ ; $i = 1$ ;**Répéter****Si**  $[R_i^{K_0}]_{\mathbb{R}^K} \in RAG^K$ Trouver  $[R_j^{K_0}]_{\mathbb{R}^K} \in A_i^K$ tel que  $Cost([R_i^{K_0}]_{\mathbb{R}^K}, [R_j^{K_0}]_{\mathbb{R}^K}) \leq Cost([R_i^{K_0}]_{\mathbb{R}^K}, [R_l^{K_0}]_{\mathbb{R}^K}), \forall [R_l^{K_0}]_{\mathbb{R}^K} \in A_i^K$ Incrémenter  $i$ ;**Tant que**  $i \leq K_0$ ; $i = 1$ ;**Répéter****Si**  $[R_i^{K_0}]_{\mathbb{R}^K} \in RAG^K$ **Si**  $Cost'([R_i^{K_0}]_{\mathbb{R}^K}, [R_j^{K_0}]_{\mathbb{R}^K}) < Th_{Cost}$ Fusionner  $[R_i^{K_0}]_{\mathbb{R}^K}$  et  $[R_j^{K_0}]_{\mathbb{R}^K}$ ; $K = K - 1$ ;Incrémenter  $Nb_{fusions}$ ;**fin Si**;**fin Si**;Incrémenter  $i$ ;**Tant que**  $i \leq K_0$ ;**Tant que**  $Nb_{fusions\_prec} < Nb_{fusions}$ ;

*s'avère ainsi multirésolution.* Typiquement, nous utilisons trois seuils, mais le choix du nombre de niveaux de résolution reste entièrement libre.

**3.5 Elimination des petites composantes.**

Compte tenu de la pondération par la taille des composantes, la segmentation proposée n'engendre pas de sur-segmentation sur les zones contours. On peut même observer une relative stabilité du nombre de régions obtenues pour un seuil de segmentation identique appliqué sur des images de complexités très différentes. L'explication réside simplement dans le fait que plus une image est complexe, plus elle va engendrer dans sa partition initiale de petites régions qui ultérieurement auront une plus forte tendance à fusionner. Néanmoins, de petites régions peuvent persister si elles présentent une forte différence avec leurs régions voisines. En conséquence, nous avons également mis en place un procédé classique de fusion des petites régions. Son seul paramètre tient en une valeur de surface, en dessous de laquelle toute région fusionnera avec la

région la plus proche d'un point de vue distance.

### 3.6 Complexité de la segmentation

La prochaine section fournira plusieurs exemples d'images segmentées, permettant d'apprécier les performances qualitatives de la méthode. Du point de vue de la complexité, l'algorithme de fusion 2.1 converge rapidement (en général de 5 à 8 itérations). Le calcul de la distance  $Cost_{Gr}$  représente à lui seul pratiquement la moitié du temps total de calcul. L'implantation des algorithmes n'a pas pour l'instant été optimisée. Les performances de segmentation d'une image  $512 \times 512$  avec 20000 blocs dans la partition initiale, en intégrant la notion de distance gradient, sont d'environ une seconde sur un PC à 2 GHz.

## 4 Applications de la segmentation

### 4.1 Codage par ROI de la texture locale

Une application directe de la représentation en régions auto-extractibles est la conception d'un schéma de codage de régions d'intérêt. A partir de la carte de segmentation disponible à la fois au codeur et au décodeur, l'un ou l'autre peut définir sa propre ROI<sup>1</sup> comme ensemble de régions de la description. La méthode procure alors tout à la fois **un outil semi-automatique de sélection des ROI**, et **probablement le moyen le plus efficace de la représenter**. En effet une ROI va alors être simplement décrite par les étiquettes des régions qui la composent.

D'autre part, la description d'une région se fonde sur la représentation en blocs. **Le rehaussement d'une ROI va donc simplement consister en l'utilisation du codeur spectral pour les blocs contenus dans la ROI**. La compatibilité entre la forme de la ROI et codage de son contenu est ainsi immédiate et totale, puisque la ROI agit directement comme une validation de type "on/off" sur les blocs à traiter.

### 4.2 Codage des images de chrominances par régions

Nous avons déjà souligné que les composantes chromatiques pouvaient être codées très efficacement avec le codeur spatial du LAR. Par ailleurs, il est clair que l'information couleur permet d'améliorer très sensiblement la segmentation des images, même si cette dernière s'opère le plus souvent dans des espaces couleur R : V : B ou L : a : b [Tre98]. Le schéma global de codage le plus naturel est donc de coder les images Y : Cr : Cb à travers le codeur spatial, puis de les transmettre au décodeur pour réaliser la segmentation sur les trois composantes. Notre but n'est pas ici d'établir une comparaison de notre segmentation avec d'autres approches dédiées. Nous avons simplement vérifié que ce mode opératoire est valide, et fournit de très bonnes descriptions en régions.

Le travail exposé ici consiste donc en l'utilisation de la représentation en régions issue de la seule composante  $LR_Y$ , afin de développer un codage par régions des deux composantes de

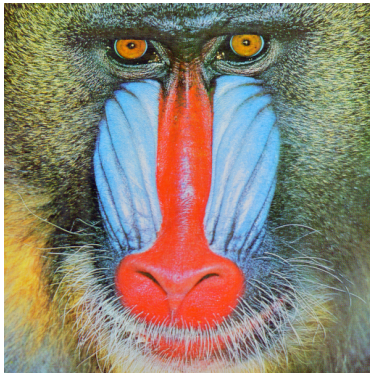
---

<sup>1</sup>ROI : Region Of Interest (région d'intérêt)

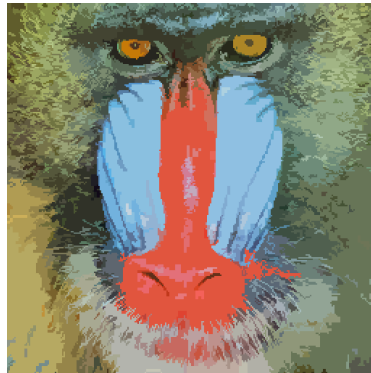
chrominance. L'ordonnancement du procédé est le suivant : à partir du codage spatial de la composante de luminance  $Y$ , la représentation en régions est obtenue au codeur et décodeur à travers un procédé commun de segmentation. Il suffit ensuite de transmettre les valeurs chromatiques moyennes contenues dans ces régions. En ajoutant une phase de codage différentiel de ces valeurs puis une phase de quantification, le coût final pour chaque région s'avère de l'ordre de 4 bits. Ainsi, le coût de codage en termes d'octets, pour les deux composantes chromatiques, correspond approximativement au nombre de régions.

Nous donnons figure 2.1 plusieurs exemples d'images reconstruites avec  $Y$  source (non codée) et  $Cr/Cb$  codées régions. Afin de mieux rapprocher ces résultats de l'image source, la représentation en régions est donnée dans l'espace couleur (moyenne dans les régions pour chaque composante de couleur).

Il faut surtout souligner ici l'étonnant rapport qualité / taux de compression sur les composantes de chrominance : le coût de leur codage se situe tout au plus en centième de bit par point et devient insignifiant. On peut par ailleurs remarquer que les défauts visibles et gênants restent dus à des problèmes initiaux de segmentation, issus d'une partie d'objet affectée à une autre région, ou plus globalement d'une fusion de plusieurs régions (fig. 2.1).



a. Image Baboon, Cr/Cb 16 bpp



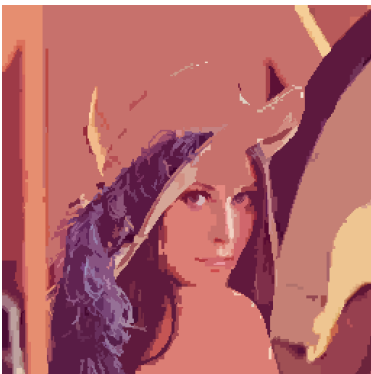
b. 671 régions



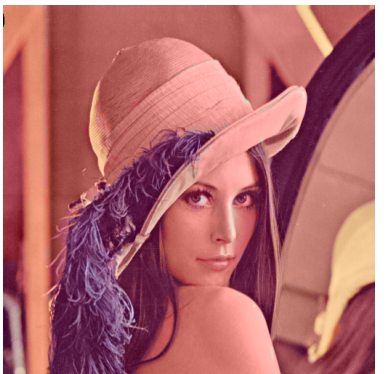
c. Cr+Cb codées région : 0.020 bpp



d. Image Lena, Cr/Cb 16 bpp



e. 308 régions



e. Cr+Cb codées région : 0.009 bpp

FIG. 2.1: Exemples d'images reconstruites avec codage des chrominances par une représentation en régions, pour un seuil de distance  $Th_{Cost} = 50$

Le codage de l'erreur commise sur les composantes chromatiques s'avère alors très simple : il consiste en effet uniquement à coder l'erreur au niveau des blocs, à travers le codeur spatial, et ce sur toute l'image ou bien simplement limité à une ROI.

## 5 Segmentation supervisée par le contrôle de la chrominance

Pour combiner à la fois compression par régions de Cr/Cb et amélioration de la qualité de segmentation par l'information couleur, nous avons défini un procédé de codage avancé, par le principe du "contrôle de la chrominance". L'idée générale est la suivante : partant toujours d'une transmission préalable de l'image Y basse résolution des luminances, le procédé de segmentation de fusion continue d'être globalement dirigé suivant un critère de luminance, mais **supervisé au codeur par un critère complémentaire de chrominance validant chaque tentative de fusion**. Le résultat de ce contrôle de supervision doit bien sûr être transmis au décodeur. La forte corrélation existant entre la composante de luminance et de chrominance, d'un point de vue structuration des objets dans l'image, induit une faible quantité d'information supplémentaire à transmettre. Le schéma bloc du codeur est donné figure 2.2.

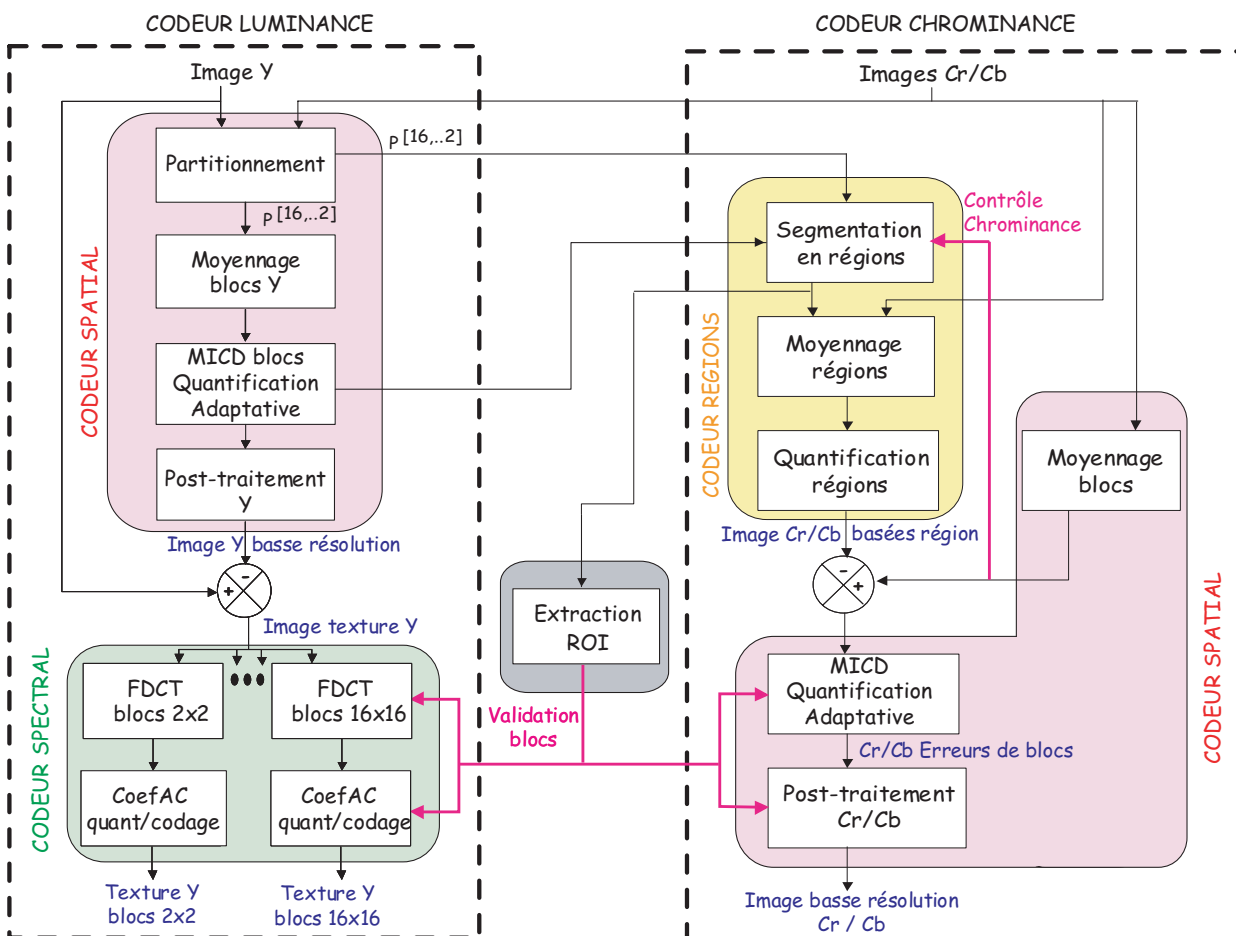


FIG. 2.2: Schéma bloc du codeur LAR couleur fondé sur une segmentation supervisée

### 5.1 Description de l'algorithme

L'algorithme de fusion, à savoir la recherche de la région la plus proche uniquement pour l'image Y, n'est pas modifié. Soient  $ControlChrom\left(\left[R_i^{K_0}\right]_{\mathbb{R}^K}\right)$  une **information binaire** à transmettre pour chaque essai de fusion, et  $CoefChrom$  un coefficient multiplicatif appliqué sur  $Th_{Cost}$ . Le critère de fusion devient (algorithme 2.2) :

---

**Algorithme 2.2** Critère de fusion : contrôle par l'information de chrominance

---

```

Si  $Cost'\left(\left[R_i^{K_0}\right]_{\mathbb{R}^K}, \left[R_j^{K_0}\right]_{\mathbb{R}^K}\right) < Th_{Cost}$ 
    Si  $CostCr'_M\left(\left[R_i^{K_0}\right]_{\mathbb{R}^K}, \left[R_j^{K_0}\right]_{\mathbb{R}^K}\right) < CoefChrom.Th_{Cost}$ 
    et  $CostCb'_M\left(\left[R_i^{K_0}\right]_{\mathbb{R}^K}, \left[R_j^{K_0}\right]_{\mathbb{R}^K}\right) < CoefChrom.Th_{Cost}$ 
         $ControlChrom\left(\left[R_i^{K_0}\right]_{\mathbb{R}^K}, \left[R_j^{K_0}\right]_{\mathbb{R}^K}\right) = 1;$ 
        Fusionner  $\left[R_i^{K_0}\right]_{\mathbb{R}^K}$  et  $\left[R_j^{K_0}\right]_{\mathbb{R}^K};$ 
    sinon
         $ControlChrom\left(\left[R_i^{K_0}\right]_{\mathbb{R}^K}, \left[R_j^{K_0}\right]_{\mathbb{R}^K}\right) = 0;$ 
    fin Si
fin Si
fin Si

```

---

Notons dans la formule précédente que nous avons uniquement considéré un calcul s'appuyant sur la moyenne des régions pour les deux composantes chromatiques. Ceci se justifie en premier lieu par l'objectif visé, à savoir une segmentation efficace pour coder Cr/Cb par régions, en second lieu par le souci de ne pas trop compliquer l'algorithme.

Par ailleurs,  $ControlChrom$ , en raison de sa très faible entropie, peut être fortement compressé par un codeur arithmétique.

D'autre part, comme un coefficient multiplicatif  $CoefChrom$  est appliqué sur  $Th_{Cost}$ ,  $Th_{Cost}$  garde alors son rôle de paramètre global de simplification de l'image.  $CoefChrom$  agit donc véritablement comme le contrôle adapté des chrominances. Il peut être ainsi vu comme un curseur de réglage entre coût du contrôle et qualité de la segmentation d'un point de vue couleur :

- si  $CoefChrom \rightarrow \infty$ , alors toutes les valeurs de contrôle valent 1, le coût entropique associé est donc nul, et la segmentation s'opère uniquement sur la luminance,
- si  $CoefChrom = 1$ , alors le seuil est le même pour les trois composantes, et le contrôle par la chrominance intervient avec une meilleure pondération,
- si  $CoefChrom < 1$ , alors la segmentation devient fortement dépendante des chrominances, mais le coût de contrôle croît avec l'entropie des valeurs binaires.

En pratique, une valeur de 0.5 assure une très bonne qualité de segmentation couleur. Néanmoins, même lorsque le contrôle est important, le coût résultant demeure très faible, comme le montrent les résultats donnés du tableau 2.1.



Th <sub>Cost</sub>	Niveau 3 : Th <sub>Cost</sub> = 100			Niveau 2 : Th <sub>Cost</sub> = 50			Niveau 1 : Th <sub>Cost</sub> = 25		
CoefChrom	2.0	1.0	0.5	2.0	1.0	0.5	2.0	1.0	0.5
Nb régions	50	81	189	178	201	304	692	713	800
Contrôle (bpp)	0.0037	0.0090	0.0310	0.0026	0.006	0.0220	0.0016	0.0045	0.0150
Régions Cr/Cb (bpp)	0.0030	0.0043	0.0089	0.0070	0.0075	0.0130	0.0240	0.0260	0.0290
Coût total (bpp) contrôle + régions	0.0067	0.0133	0.0399	0.0096	0.0135	0.0350	0.0256	0.0305	0.0440
PSNR Cr/Cb (dB)	32.15	34.3	35.92	33.42	34.8	36.2	34.76	35.3	36.86
Erreurs sur Cr/Cb Codeur spatial (bpp)	0.08	0.072	0.061	0.072	0.070	0.067	0.063	0.070	0.056
Coût total Cr/Cb Niveau blocs (bpp)	0.0867	0.0853	0.1000	0.0816	0.0835	0.1020	0.0886	0.1005	0.1000

TAB. 2.1: Influence des paramètres de segmentation sur “Barbara”.

La décision de fusion doit évidemment être accompagnée d’un marquage de la région rejetée, de manière à ce que celle-ci ne reste pas la meilleure candidate pour la fusion d’un point de vue luminance. Nous avons donc introduit un graphe de dissymétrie des régions ou RDG (Regions Dissymmetric Graph), sous-graphe du RAG. L’ensemble  $D_i^K$  associé fournit par conséquent l’ensemble des classes rejetées pour une classe donnée. Sa gestion lors de la fusion de deux régions est analogue à celle du RAG.

## 5.2 Résultats et influence des paramètres de segmentation.

Pour juger de l’influence des paramètres  $Th_{Cost}$  d’une part, et  $CoefChrom$  d’autre part, à la fois en termes de qualité de segmentation et de coût de codage des images de chrominance, nous avons fait varier ces deux paramètres indépendamment. La taille de l’image “Barbara” utilisée est de 512x512 pixels codés sur 24 bits, le nombre de blocs initiaux dans la partition est de 20153 pour une image Y compressée à 0.19 bpp, et le coût codage direct des composantes Cr et Cb par le codeur spatial est de 0.12 bpp avec un PSNR de 40.1 dB. Les résultats quantitatifs sont résumés dans le tableau 2.1, et les images de segmentation ainsi reconstruites sont présentées figure 2.3. D’autres exemples de segmentation sont également proposés figure 2.4.

Plusieurs remarques peuvent être faites sur les résultats.

- Le codage des images de chrominance par régions permet d’atteindre des forts taux de compression.
- L’influence du paramètre  $CoefChrom$  sur le nombre de régions générées est plus forte pour les partitions contenant peu de régions. Le coût du contrôle s’accroît. En revanche avec beaucoup de régions, son coût reste faible comparé à celui du codage des régions chromatiques, mais il apporte toujours un gain significatif du point de vue distorsion.
- Lorsque l’on code ensuite l’erreur des images de chrominance au niveau des blocs dans la partition, c’est à dire par le codeur spatial, les résultats sont relativement stables, et assez proches d’un codage direct (0.12 bpp).

La figure 2.5 fournit des exemples d’images reconstruites avec la composante Y originale, et les composantes Cr/Cb codées régions, pour différents taux de compression. On peut constater qu’un codage chromatique avec un nombre suffisant de régions, permet d’obtenir une qualité acceptable de reconstruction dans un contexte bas débit, évitant ainsi le codage de l’erreur par le codeur spatial. On notera également la relative stabilité du nombre de régions résultantes pour

différentes images.



FIG. 2.3: Résultats de segmentation en fonction des paramètres  $Th_{Cost}$  et  $CoefChrom$  sur image “Barbara”.

### 5.3 Codage par régions contraintes par masques

Cette technique de contrôle par la chrominance a été étendue très simplement pour contraindre la segmentation à toute forme binaire d’objet donné. Cette solution s’avère par exemple nécessaire lorsque la ROI est déjà définie au codeur de manière manuelle ou automatique (principe du VOP de forme quelconque dans MPEG-4). Le partitionnement initial de l’image inclut alors le masque binaire de l’objet. La segmentation s’effectue alors sous le contrôle de la forme, afin de garantir des régions soit entièrement externes, soit entièrement internes à l’objet. Celui-ci est alors désigné par l’ensemble des régions qui le composent. Le procédé est illustré figure 2.6, avec un rehaussement de qualité appliqué uniquement dans l’objet.





FIG. 2.4: Résultats de segmentation sur “Baboon”, “peppers”, “Fruits” et “Lena” ( $CoefChrom = 0.5$ ).

## 6 Conclusion

Dans ce chapitre, nous avons introduit une représentation en régions s’appuyant sur l’information basse résolution produite par le codeur spatial du LAR. Les caractéristiques de cette image bas débit, en particulier son rendu intégral des contours, sont exploitées afin de réaliser une segmentation efficace. Le résultat fournit une **description hiérarchique** de l’image en régions, **de bonne qualité** d’un point de vue représentation de la scène, et surtout **sans coût supplémentaire de description des formes**.

Un atout important de notre schéma de représentation tient dans sa partition initiale en blocs, à la base des régions formées. Il s’ensuit une **adéquation parfaite entre forme et contenu des régions**. Une application immédiate réside dans le codage par régions d’intérêt dans l’image : la description de celles-ci entraîne alors un coût quasi-nul, et le contenu d’une région d’intérêt donnée étant directement et simplement rehaussé soit à l’aide du codeur spectral, soit via l’utilisation d’une des méthodes pyramidales (chapitres 5, 6 et 7) pour un codage localement sans perte.

L’introduction d’un **contrôle de la segmentation par l’information chromatique** permet d’obtenir un schéma de codage très efficace tant du point de vue taux de compression que de la qualité des images reconstruites. L’introduction du codage par régions des images de chrominance autorise en effet des taux de compression inégaux, par comparaison aux techniques classiques.

Cette technique de segmentation va, en outre, servir de point de départ aux réflexions menées sur l’extension de la méthode LAR à des problématiques liées à la vidéo (voir chapitre 10). En effet, si il existe déjà des méthodes de compression vidéo intégrant la notion de régions, elles se révèlent souvent limitées par un coût important dû à la description des formes. Notre algorithme, s’affranchissant de ce problème, semble une solution intéressante pour le codage de séquences d’images à très bas débits. Une question demeure : comment se comporte notre segmentation si elle s’applique non plus sur une seule image, mais plutôt sur une succession d’images ? Le chapitre



a) "Barbara", Cr+Cb : 16 bpp



b) Cr+Cb codée codeur spatial :  
0.12 bpp, taux compression = 133



c) Cr+Cb codée avec 800 régions,  
 $Th_{Cost} = 25$ ,  $CoefChrom = 0.5$  :  
0.044 bpp, taux compression = 364



d) Cr+Cb codée avec 304 régions,  
 $Th_{Cost} = 50$ ,  $CoefChrom = 0.5$  :  
0.035 bpp, taux compression = 457



e) Cr+Cb codée avec 81 régions,  
 $Th_{Cost} = 100$ ,  $CoefChrom = 1$  :  
0.0043 bpp, taux compression = 3720



f) Cr+Cb codée avec 50 régions,  
 $Th_{Cost} = 100$ ,  $CoefChrom = 2$  :  
0.0030 bpp, taux compression = 5333

FIG. 2.5: Codage des chrominances par régions



*JPEG2000 : 0.17 bpp*



*LAR : 0.17 bpp (Quadtree : 0.03  
Y : 0.11, Cr/Cb Reg : 0.03)*



*Masque binaire ROI*



*90 Régions (21 dans ROI)*



*ROI rehaussée seule : 0.68 bpp*



*Basse résolution + ROI : 0.22 bpp*

FIG. 2.6: Codage par ROI sur image “Akiyo” avec contraintes de chrominance et masque binaire.

10 présentera une première évaluation de cette extension à la video, avant de présenter quelques idées de réalisations futures.

## Deuxième partie

# Compression sans perte d'images fixes par la méthode LAR : approches pyramidales



## Chapitre 3

# Décomposition Pyramidale : les différentes approches

Si, par le passé, on s'est efforcé en priorité d'améliorer le taux de compression d'un schéma donné, on s'attache en plus désormais à ajouter de nouvelles fonctionnalités. En particulier, la transmission ou le décodage progressifs des données ouvrent de nombreuses perspectives d'applications nouvelles. La navigation de données en est typiquement un exemple : le flux binaire global stocké doit permettre la reconstruction d'une version grossière de l'image par le traitement d'une simple partie de ce flux.

Ce chapitre décrit le principe général de la décomposition multirésolution d'une image, et des algorithmes pyramidaux en particulier. Il est organisé comme suit : au préalable, les notions de "embedded coding", progressivité et scalabilité sont définies. Dans un deuxième temps, la théorie de l'analyse multirésolution (AMR) est exposée dans ses grandes lignes, avant d'aborder le concept de la décomposition en sous-bandes. Ceci nous amène tout naturellement à étudier les principes généraux de construction des ondelettes (§4). Enfin, la section 5 s'attache à décrire les différents schémas de construction d'une représentation pyramidale d'une image.

## 1 Définitions préalables des notions de base

### 1.1 "Embedded Coding"

Shapiro donna la définition fondamentale suivante [Sha93] : un codage dit *embedded*<sup>1</sup> est un système dans lequel le flux codé est ordonné par importance visuelle, ou, de façon plus générale, ordonné suivant un critère de qualité. Les figures 3.1 et 3.2 illustrent la différence entre la transmission d'un flux codé classiquement, et un flux codé du type *embedded*.

Une définition stricte du codage *embedded* est la suivante : si deux fichiers produits par un encodeur suivant un critère de qualité donné sont de tailles respectives  $M$  et  $N$  bits, avec  $M > N$ , alors le fichier de taille  $N$  est identique aux premiers  $N$  bits contenus dans le fichier de taille  $M$ .

De tels schémas possèdent les avantages suivants [Dav98].

- Il est possible d'obtenir au décodage une image dont la précision augmente progressivement.

Dans le cas d'une pyramide, ceci est facilement réalisé : il suffit d'envoyer les coefficients

---

<sup>1</sup>Embedded : le terme n'a pas de réel équivalent en langue française. Nous pourrions par exemple le traduire par "emboîté".

transformés de la résolution la plus basse ( $L_{max}$ ), puis de raffiner l'image par l'envoi successif des coefficients correspondant à des niveaux de résolution supérieure.

- L'encodage peut atteindre des débits précis.
- Le décodeur peut cesser de récupérer les données à tout moment, et reconstruire alors une image de la meilleure qualité possible avec le nombre de bits décodés disponibles. Cette propriété est intéressante dans des applications de type "broadcast" (diffusion des données) où plusieurs décodeurs, de capacité de calcul, de bande passante et d'affichage différents accèdent à la même ressource.

Enfin, il est à noter que les codes *embedded* trouvent des applications dans les domaines de l'indexation et la navigation de données.

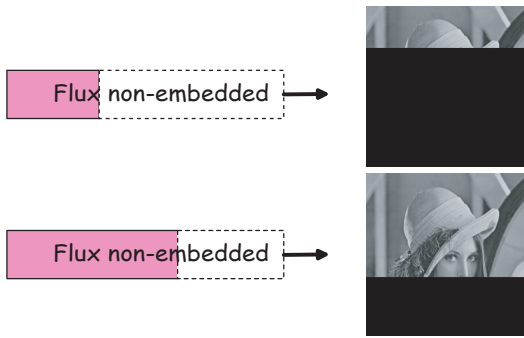


FIG. 3.1: Transmission d'un codage *non-embedded*.

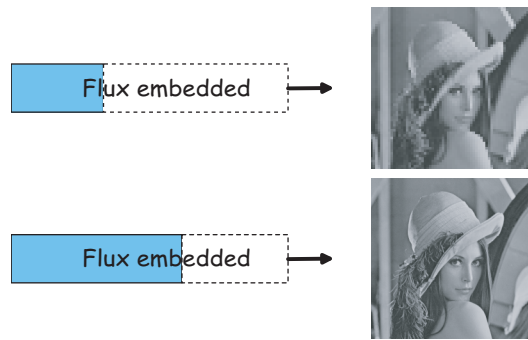


FIG. 3.2: Transmission d'un codage *embedded*.

## 1.2 Scalabilité<sup>2</sup> et progressivité

La "scalabilité" est un concept fondamental de l'algorithme JPEG-2000, défini quant à lui au chapitre 4, §4. Dans la littérature, la scalabilité désigne bien plus que le simple caractère *embedded* du codeur, défini par la capacité à représenter une image donnée en une variété d'images de tailles potentiellement différentes et à des taux de compression variables. La scalabilité est une propriété intrinsèque du flux de données. Taubman et Marcellin donnent la définition suivante [TM02] :

**Définition 3.1 (Scalabilité)** *Un flux de données compressées est dit scalable si il consiste en une collection de flux plus petits imbriqués, chacun de ces sous-flux représentant une compression efficace de l'image originale, ou d'une portion de celle-ci.*

Dans ce même article, Taubman et Marcellin proposent de définir plusieurs types de scalabilité pour une même méthode de compression donnée, à savoir : scalabilité par résolution, distorsion, Région d'Intérêt, composante et scalabilité temporelle.

Ainsi, une représentation compressée de l'image est **scalable en termes de résolution** (scalabilité spatiale) si elle contient des sous-ensembles identifiables qui codent des versions successives de plus basse résolution de l'image source. Elle peut être aussi **scalable en termes de distorsion** si elle contient des sous-ensembles identifiables qui codent des images successives de qualité croissante, pour une pleine résolution. De ce fait, si un flux de données possède ces deux précédentes

<sup>2</sup>Scalabilité : il n'existe pas en français de terme équivalent au mot "scalability". "Scalabilité" ou encore "scalable" sont désormais couramment admis dans la communauté des traiteurs du signal.



propriétés, il est possible de reconstruire l'image au décodeur à des niveaux de résolution différents pour une qualité visuelle variable. La figure 3.3 illustre l'imbrication des sous-flux scalables en termes à la fois de distorsion et de résolution.

Notons par ailleurs que si les décompositions multirésolution (voir §2) constituent la clé de voûte d'une approche scalable par résolution, le codage "embedded" est fondamental pour la scalabilité en termes de qualité visuelle de l'image reconstruite.

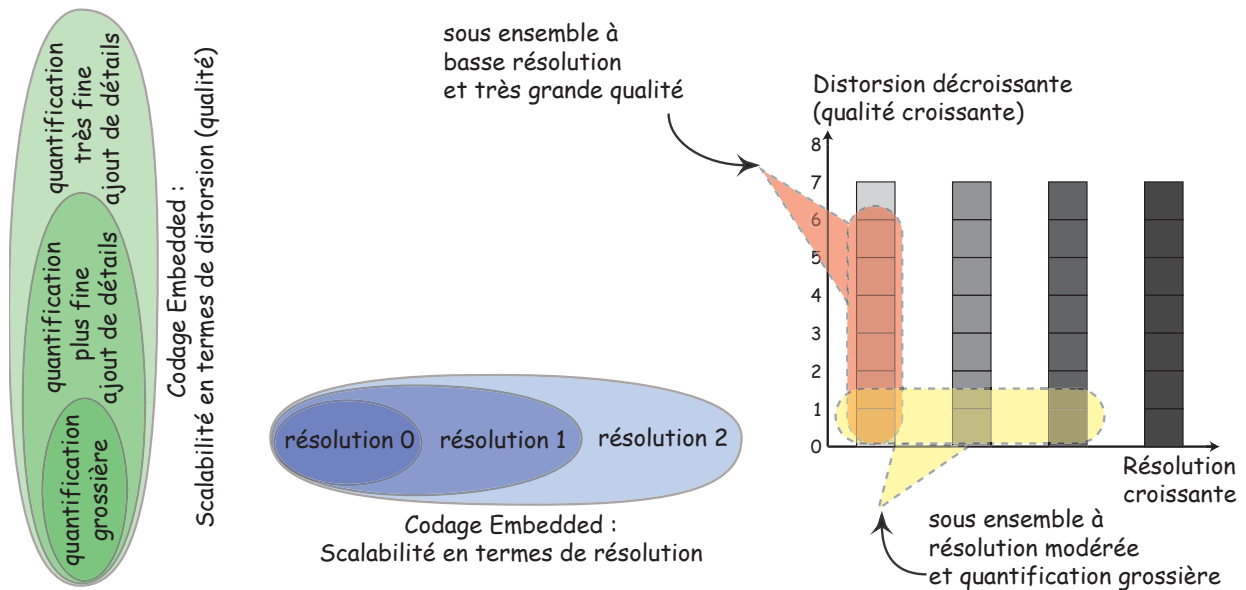


FIG. 3.3: Visualisation des différents types possibles de scalabilité pour un flux de données compressées.

De plus, deux autres opérations peuvent être qualifiées de scalable : l'extraction d'un sous-flux codant pour une **région d'intérêt**, et la prise en compte différenciée d'une ou plusieurs **composantes**<sup>3</sup> de l'image. Enfin, dans le cas de codage de séquences d'images, la **scalabilité temporelle** consiste en l'accès aléatoire aux images de la séquence, ou inversement, à l'insertion d'images à la même résolution dans le flux vidéo, afin d'augmenter le débit vidéo (figure 3.4). Les standards MPEG2 et MPEG4 associent cette notion à la reconstruction d'une séquence à débit minimal [CR03].

La notion de **progressivité** est généralement employée pour caractériser la fonction de *transmission* de l'information codant l'image [IP99]. La progressivité se définit donc comme la capacité d'une chaîne de codage à transmettre un flux de données organisé de telle sorte que le décodeur puisse reconstruire par étapes successives des images de qualité et de taille variables. La première version de l'image, rapidement disponible, correspond à une version grossière de l'image originale. Les passes suivantes contribuent à raffiner l'image initialement reconstruite. Bien sûr, de par la transmission progressive des données, l'envoi des informations ligne par ligne n'est plus envisageable.

<sup>3</sup>Composante : le terme, générique, désigne ici non seulement les différents canaux codant pour la couleur, comme la luminance, et la chrominance, mais aussi les métadonnées (texte, graphique...) qui sont présents dans certains documents comme les images médicales.



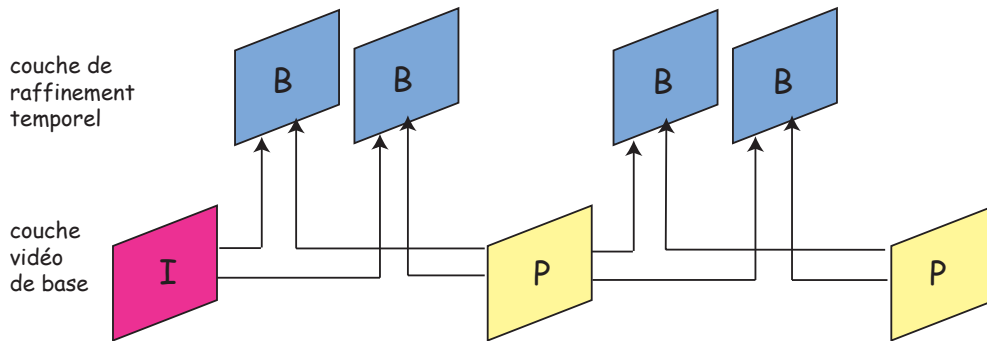


FIG. 3.4: Scalabilité temporelle pour MPEG : un exemple. Image I : image codée intra (opérations effectuées sur l'image seule) - Image P : image codée par prédiction temporelle "forward" - Image B : image codée par prédiction temporelle bidirectionnelle.

Pour une information complète sur la progressivité, le lecteur pourra se reporter à la thèse écrite par Nicolas de Maistre [dM98].

## 2 Analyse multirésolution

Le multirésolution est une notion récurrente dans l'état-de-l'art des méthodes de traitement du signal. Cette section se propose de définir l'analyse multirésolution, puis d'en extraire les propriétés qui ont conduit à son intense utilisation.

### 2.1 Définition intuitive

Une analyse multirésolution décompose le signal original en une suite d'images sur une gamme très étendue d'échelles : cette opération peut s'apparenter à la cartographie. Lors du parcours descendant de la représentation multiéchelle de l'image (de la plus grossière à la plus fine), on accède à des informations de plus en plus précises du signal donné. L'analyse s'effectue en observant ce qui diffère d'une échelle à l'autre, c'est-à-dire la différence d'informations entre deux représentations consécutives, ou encore les détails caractérisant une résolution donnée.

La multirésolution spatiale engendre une progressivité de la taille de l'image, et donc de la résolution. Une représentation multirésolution constitue de ce fait une série d'images représentant une même scène observée à différentes résolutions spatiales. L'empilement particulier de cette série d'images "dérivées" permet de construire une **pyramide**, composée d'une succession de niveaux, dont le plus bas est l'image d'origine, et le plus haut, une version grossière et réduite en taille de cette image.

### 2.2 Cadre théorique

L'analyse multirésolution (AMR) est issue d'une théorie plus globale, et plus moderne : l'analyse en ondelettes. Le travail de doctorat de Mallat (dirigé par Meyer), commencé en 1986, aboutit à la mise en place de la théorie générale sur l'AMR [Mal89a]. Suite à cet article, de nombreuses

méthodes fondées sur l'exploitation de l'AMR ont vu le jour. La rédaction de cette sous-section s'appuie essentiellement sur les références suivantes : [MMOP03], [BGM<sup>+</sup>00] et [Pag03].

Soit  $f$  le signal à analyser, et  $L^2(\mathbb{R})$  l'espace des fonctions de  $\mathbb{R}$  dans  $\mathbb{R}$  de carré intégrable. Une *analyse* (orthogonale) *multirésolution* de  $L^2(\mathbb{R})$  est une famille  $M = \{V_j\}_{j \in \mathbb{Z}}$  de sous-espaces vectoriels de  $L^2(\mathbb{R})$  fermés emboîtés, ce que l'on peut réécrire par

$$V_\infty \dots \subset V_1 \subset V_0 \subset V_{-1} \dots \subset V_{-\infty}. \quad (3.1)$$

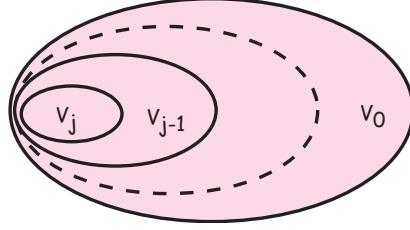


FIG. 3.5: Sous-espaces vectoriels d'approximation emboîtés.

La figure 3.5 illustre la relation précédente. Chaque sous-espace constitue l'ensemble de toutes les approximations possibles d'un même signal à l'échelle associée au sous-espace considéré. Les propriétés suivantes sont fondamentales. Tout d'abord,

$$V_j \text{ est un sous-espace fermé de } L^2(\mathbb{R}). \quad (3.2)$$

Cette précédente propriété garantit l'existence d'une projection orthogonale de  $f$  sur chaque sous-espace  $V_j$ .

De plus, on peut écrire que

$$\bigcap_{j \in \mathbb{Z}} V_j = 0. \quad (3.3)$$

Autrement dit, quand  $j \rightarrow \infty$ ,  $V_j$  est réduit à la fonction nulle. L'équation 3.3 montre qu'à résolution minimale, l'image est entièrement perdue.

L'union des sous-espaces est dense dans  $L^2(\mathbb{R})$  (équation 3.4) : ceci assure que la suite  $M = \{V_j\}_{j \in \mathbb{Z}}$  converge dans  $L^2(\mathbb{R})$ , et donc que la suite des projections orthogonales converge vers  $f$ . Ainsi,

$$\overline{\bigcup_{j \in \mathbb{Z}} V_j} = L^2(\mathbb{R}), \quad (3.4)$$

où  $\overline{(\circ)}$  désigne l'adhérence<sup>4</sup> de  $(\circ)$ .

Par ailleurs, tous les espaces  $V_j$  résultent d'une *dilatation* ou une *contraction* dyadique des fonctions d'un espace unique :

$$\forall j \in \mathbb{Z}, v(t) \in V_j \Leftrightarrow v(2t) \in V_{j-1}, \quad (3.5)$$

qui nous donne :  $v(t) \in V_0 \Leftrightarrow v(2^j t) \in V_j$ .

<sup>4</sup>L'adhérence d'un espace désigne le plus petit espace fermé contenant cet espace.

Enfin, la propriété de translation de fonctions indique :

$$\exists g \in V_0 \quad \text{telle que} \quad \{g(t - k)\}_{k \in \mathbb{Z}} \text{ est une base de Riesz de } V_0, \quad (3.6)$$

où une base de Riesz se définit de la façon suivante :  $\{e_k\}_{k \in \mathbb{Z}} \subset L^2$  est une base de Riesz de  $L^2$  si et seulement si  $\forall h \in L^2, \exists ! \alpha \in l^2(\mathbb{Z})$  tel que  $h = \sum_{k \in \mathbb{Z}} \alpha_k e_k$ .  $l^2(\mathbb{Z})$  constitue l'espace des suites de  $\mathbb{Z}$  dans  $\mathbb{Z}$  de carré intégrable.

Une **fonction d'échelle**  $\varphi$ , ou fonction d'interpolation, associée à l'ensemble de ses translatées entières engendre une base orthonormée de  $V_0$ .

## 2.3 Orthogonalité et biorthogonalité

Lorsque la base de Riesz (voir paragraphe précédent) est une base orthogonale, l'approximation multirésolution est orthogonale, et la fonction engendrant cette base (autrement dit l'atome de base) est appelé fonction d'échelle. Il est toujours possible d'orthogonaliser, -c'est-à-dire appliquer des translations entières d'une fonction élémentaire sur la base considérée-, une approximation multirésolution.

L'orthogonalité impose cependant des contraintes qui peuvent nuire à sa construction. L'une des plus importantes est la suivante : une fonction d'échelle  $f$  orthogonale à support compact ne peut être à la fois symétrique et continue. La propriété de symétrie est néanmoins utile à l'analyse des signaux de longueur finie.

Certaines de ces restrictions (notamment sur la symétrie) peuvent être levées en utilisant des approximations multirésolutions biorthogonales, afin de construire des ondelettes aux propriétés plus attractives : deux ondelettes  $\varphi$  et  $\tilde{\varphi}$  sont cette fois-ci élaborées en dualité. Le cas orthogonal revient à écrire  $\varphi = \tilde{\varphi}$ .

La biorthogonalité sera de nouveau évoquée au paragraphe 4.3.1.

## 2.4 Pourquoi une décomposition multirésolution ?

Les avantages des techniques multirésolution ont été décrites par Mallat [Mal89a] en particulier. Les points importants sont les suivants :

- l'analyse opérée par le système visuel humain s'appuie sur le contenu fréquentiel de l'image. L'image perçue par la rétine est décomposée par certaines cellules qui agissent comme des filtres passe-bandes de largeur égale à une octave, d'orientation spatiale et de fréquences déterminées [SSB93]. Une décomposition dyadique pyramidale de l'image, où chaque niveau correspond à la résolution moitié du niveau précédent, imite cette propriété.
- La multirésolution pour le codage d'images possède deux avantages majeurs. D'une part, la structure hiérarchique autorise le traitement des imagelettes successives par des méthodes différentes : typiquement, la quantification s'adapte à l'échelle à laquelle elle est appliquée. D'autre part, la multirésolution constitue un dispositif "approche-précision", où les éléments grossiers donnent une indication sur le moyen de traiter des structures fines de l'image.
- Étant donnée la similarité des images aux différents niveaux, la décomposition hiérarchique offre une propriété intéressante de corrélation spatiale (et décorrélation fréquentielle) des

données : cela signifie qu'il est possible de prédire l'information à petite échelle à l'aide des informations issues des échelles supérieures.

La décomposition en sous-bandes, présentée dans la section suivante, provient de l'extension en 2D des travaux de Crochiere en 1976 [CWF76], qui utilisa ce concept pour le codage de la voix.

### 3 La décomposition en sous-bandes

Le codage en sous-bandes (SBC)<sup>5</sup> est tout particulièrement adapté aux applications multimedia, où les notions de "scalabilité" et de "dégradations tolérables" pour l'œil humain sont importantes. Du point de vue du codage, cela signifie :

- la version dite "basse résolution" de l'image transmise doit être décodée avec le minimum d'information possible,
- l'image pleine résolution ne sera décodée seulement si l'utilisateur le désire.

La décomposition en sous-bandes est devenue incontournable pour une représentation multirésolution et une transmission progressive des données.

Les principes généraux de construction des sous-bandes sont exposés ci-après, ainsi que les propriétés inhérentes à cette décomposition.

#### 3.1 Principes généraux

Le concept fondamental du SBC est de séparer le spectre de l'image en plusieurs bandes de fréquences, puis d'encoder celles-ci au moyen d'un codeur et d'un débit adaptés aux statistiques de la bande. Les transformations sous-bandes font majoritairement partie de la famille des transformations linéaires.

Le schéma de codage consiste généralement en l'application d'un jeu de filtres passe-bas/passe-haut, ou passe-bande, sur le signal d'entrée. Chaque sous-bande alors constituée contient une gamme limitée de fréquences du spectre de l'image originale, correspondant à l'information présente à une échelle spatiale donnée. Cette propriété permet de sous-échantillonner par 2 dans les deux dimensions chacune de ces sous-bandes. Cette description correspond à la partie dite **analyse** ou encore **décomposition** d'un **banc de filtres à deux canaux**.

Au décodeur (partie **synthèse** ou **reconstruction** d'un banc de filtres), les signaux relatifs à chaque bande sont décodés, sur-échantillonnés par 2 (insertion d'un zero entre chaque échantillon), filtrés, puis additionnés afin de reconstruire l'image transmise.

Les premières applications des sous-bandes au codage des images ont été élaborées par Woods et O'Neil, au moyen de bancs de filtres miroirs en quadrature (*QMF*) [WO86], séparables et unidimensionnels. Soient  $h_0$  et  $h_1$ , les filtres d'analyse (respectivement passe-bas et passe-haut) au codeur, et  $g_0$  et  $g_1$ , les filtres (respectivement passe-haut et passe-bas) présents au décodeur.

---

<sup>5</sup>SBC : SubBand Coding

Les QMF sont définis par la relation suivante :

$$\begin{cases} g_0(n) = (-1)^n h_1(n) \\ g_1(n) = (-1)^n h_0(n) \end{cases}$$

Il est à noter que la plupart des solutions de codage utilisent des transformées 1D. Le traitement des images se fait donc par l'application successive des filtres sur les lignes, puis sur les colonnes. Le maximum d'information est alors concentré sur les basses fréquences : celles-ci représentent les zones homogènes de l'image, alors que les hautes fréquences contiennent les informations de contour. Le processus peut être répété sur une ou plusieurs bandes de fréquences.

### 3.2 Propriétés de la décomposition

La décomposition en sous-bandes regroupe l'énergie à la fois dans l'espace des fréquences et dans le domaine spatial. Pour les signaux à deux dimensions, une région localisée dans l'espace fréquentiel représente une échelle particulière et une orientation donnée dans le domaine spatial. Cette spécificité de la transformée lui permet d'extraire les structures orientées d'ordre supérieur, comme les contours et les lignes [SA92]. Ces particularités peuvent être exploitées afin d'améliorer le taux de compression final, au moyen d'une quantification et d'un codage entropique des coefficients préalablement classifiés en fonction du contexte [YOY99].

Un avantage majeur du codage en sous-bandes sur les schémas de codage de type JPEG est l'absence d'effets de blocs. Cependant, le sous-échantillonnage introduit un repliement de spectre caractéristique. La plupart des filtres sont conçus pour minimiser ces distorsions, mais des effets dits de rebonds ("*ringing effects*"), dus aux phénomènes de Gibbs, sont observables sur les contours de l'image [EVE97].

Enfin, la recherche d'une décomposition en sous-bandes optimale consiste à la récupération, après traitement, de sous-bandes de spectre le plus plat possible. Ce principe, fil conducteur de l'ensemble des études menées sur ce thème, fait l'objet du paragraphe suivant.

## 4 De la décomposition en sous-bandes aux ondelettes

Les méthodes de mise en œuvre des transformées en ondelettes constituent une sous-famille des SBC. La théorie qui régit les bases d'ondelettes (§4.1) a été élaborée en complément de l'analyse multirésolution, vue au paragraphe 2. Daubechies fut la première à découvrir le lien entre les deux concepts : l'itération des QMFs sous certaines conditions pouvaient conduire à l'expression d'une transformée continue en ondelettes [Sah99]. Mallat, grâce à son algorithme rapide de décomposition pour la transformée discrète en ondelettes (§4.2), établit la relation entre les bases orthonormées d'ondelettes et les bancs de filtres classiques en traitement du signal [Mal89b]. L'émergence des techniques basées sur l'exploitation des ondelettes (en particulier le standard JPEG2000) nous incite à développer cet outil mathématique, et à présenter les schémas avancés de décomposition en ondelettes.

### 4.1 Base d'ondelettes.

Reprenons les notations et les définitions utilisées pour l'analyse multirésolution (AMR) du paragraphe 2.1 : l'espace des approximations  $y$  est spécifié.

L'espace des détails  $W_j$  correspond simplement au supplémentaire orthogonal de  $V_j$  dans  $V_{j-1}$  :

$$V_{j-1} = V_j \oplus W_j. \quad (3.7)$$

Autrement dit, c'est l'espace qui contient la différence d'information entre une approximation à la résolution  $j$ , et une approximation à la résolution  $j - 1$ . Les sous-espaces  $W_j$ , pour  $j \in \mathbb{Z}$ , sont orthogonaux entre eux.

Ces espaces de détails sont, comme les résolutions, engendrés par les translatées et les dilatées d'une unique fonction, appelée **fonction d'ondelette**. L'ordre d'approximation de la multirésolution correspond au nombre de moments nuls de l'ondelette. Il représente également la capacité de l'ondelette à détecter les singularités isolées d'un signal. La figure 3.6 présente une représentation pyramidale des données.

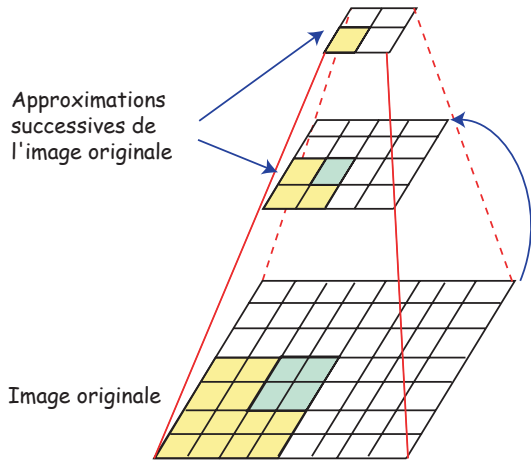


FIG. 3.6: Représentation par algorithme pyramidal.

Différence d'information  
entre deux approximations  
successives

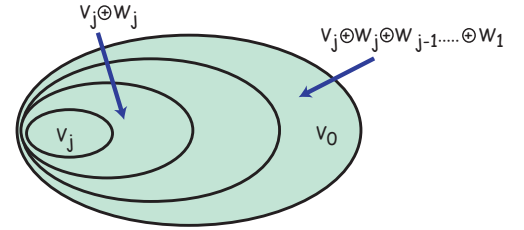


FIG. 3.7: Sous-espaces vectoriels de détails emboîtés.

### 4.2 Algorithme de Mallat : les clés de sa construction.

L'algorithme de Mallat [Mal89b] propose une solution de construction d'une décomposition au moyen d'une transformée ondelettes discrète (DWT<sup>6</sup>). Soient  $\varphi$  la fonction d'échelle et  $\phi$  l'ondelette. Pour une fonction  $f \in L^2(\mathbb{R})$  donnée, calculer sa décomposition en ondelettes revient à projeter  $f$  sur les sous-espaces  $V_j$  et  $W_j$ , munis respectivement des bases orthonormées  $\{\phi_{i,j}\}_{k \in \mathbb{Z}}$  et  $\{\psi_{i,j}\}_{k \in \mathbb{Z}}$ . Les coordonnées ainsi recueillies ( $A^j$  et  $D^j$ ) correspondent aux coefficients d'approximation et de détail, suivant :

$$\begin{aligned} A^j &= \sum_{p \in \mathbb{Z}} a_p^j \varphi_{j,p}, \\ D^j &= \sum_{p \in \mathbb{Z}} d_p^j \psi_{j,p}. \end{aligned} \quad (3.8)$$

<sup>6</sup>DWT : Discrete Wavelet Transform

L'observation du système d'équations 3.8 permet de concevoir, sous certaines conditions ( $\varphi$  à support compact), la suite  $(a_p)_{p \in \mathbb{Z}}$  comme un filtre passe-bas. De ce filtre à réponse impulsionnelle finie (FIR)  $h_0$ , on peut en déduire le filtre miroir  $h_1$  en quadrature passe-haut, nécessaire à la décomposition en ondelettes. Pour de plus amples informations, le lecteur pourra se reporter au livre de Misiti *et al.* [MMOP03]. La suite de l'algorithme consiste à itérer le processus sur le signal d'approximation (schéma figure 3.8).

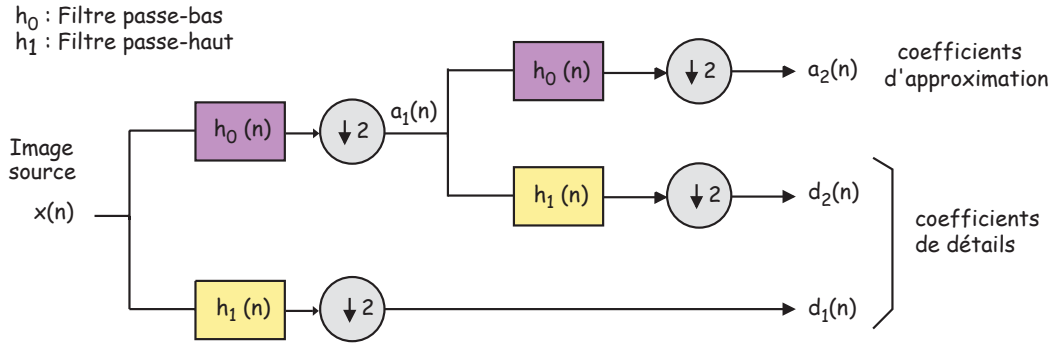


FIG. 3.8: Schéma-bloc d'une transformation ondelettes à 2 niveaux (signal 1D)

### 4.3 Extension de la construction des ondelettes

Les travaux sur les ondelettes ont conduit à des algorithmes de généralisation de la méthode de construction des transformées. Dans cette section, nous nous proposons de présenter brièvement les notions concernant les bases biorthogonales d'ondelettes, l'algorithme à trous, et enfin les paquets d'ondelettes.

#### 4.3.1 Bases biorthogonales d'ondelettes

Historiquement, les premières ondelettes construites par Mallat et Meyer [Mal89a] étaient orthogonales. Toute base d'ondelettes non orthogonale reçoit la qualification de biorthogonale (§2.3). On peut montrer par ailleurs qu'il n'existe pas (à un cas trivial près : l'ondelette de Haar) d'ondelette orthogonale symétrique. Ainsi, l'utilisation des ondelettes biorthogonales, **généralisation** de l'ondelette orthogonale, permet une reconstruction exacte du signal.

La réponse impulsionnelle des filtres QMF (filtres miroir en quadrature) suit le raisonnement suivant : les filtres de reconstruction possèdent les mêmes coefficients que les filtres de décomposition, mais disposés dans le sens inverse. Dans le cas des bancs de filtres biorthogonaux, cette propriété n'est plus vraie : les filtres de reconstruction ont un nombre et des valeurs de coefficients différents des filtres de décomposition. En fait, la construction d'ondelettes biorthogonales revient à la synthèse de **filtres à reconstruction parfaite** possédant une propriété de stabilité.

La norme JPEG 2000 (voir chapitre suivant), recommande l'utilisation des filtres biorthogonaux dits 9/7 de Daubechies ou 5/3 de Le Gall. La signification de ces nombres est la suivante : la réponse impulsionnelle du filtre passe-bas (resp. passe-haut) du banc 9/7 possède 9 (resp. 7) coefficients.

### 4.3.2 Construction d'une transformée dyadique en ondelettes : algorithme à trous

Dans le cas où l'information directionnelle importe peu, un algorithme non directionnel (contrairement à l'algorithme de Mallat) peut suffire. L'algorithme à trous répond à cette propriété. Développé par Holdschneider *et al.*, il a été notamment mis en œuvre par Bijaoui *et al.* [JF94]. La transformée en ondelettes dyadique rapide se construit alors par la conception de bancs de filtres. Cette implantation est très proche de la transformée en ondelettes (bi)orthogonales rapide, sauf qu'aucun sous-échantillonnage n'y est effectué (figure 3.9). Pour éviter ce sous-échantillonnage, un système d'entrelacement des approximations est élaboré. Ainsi, à l'échelle 1, l'ensemble des valeurs est obtenu par translation des pixels de l'image à la résolution 0. Pour le niveau 2, il suffit alors de ne pas tenir compte des pixels entrelacés (pas de 2 dans la convolution). La récupération des coefficients d'ondelettes se fait simplement par le calcul de la différence entre deux approximations successives des signaux dans la pyramide multi-résolution. Comme la technique n'inclut pas de sous-échantillonnage, la reconstruction se fait directement par l'addition de l'ensemble des plans d'ondelettes intégrant également le dernier plan d'approximation. L'avantage de l'algorithme à trou par rapport à l'algorithme de Mallat est qu'il permet le calcul de la transformée sur l'ensemble des abscisses. En revanche, cet avantage devient aussi un inconvénient puisque la relation signal original - signal transformé n'est plus bi-univoque : la reconstruction du signal original à partir de ces coefficients et signaux transformés n'est donc pas exacte.

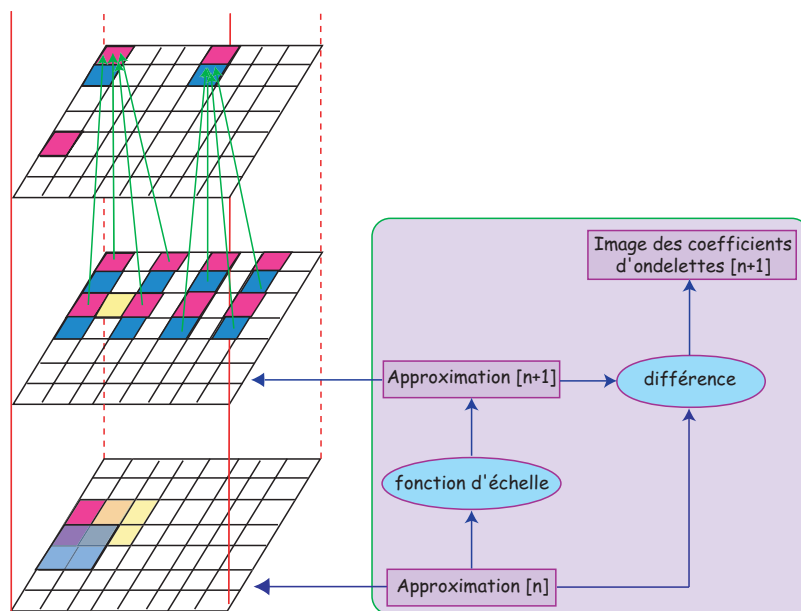


FIG. 3.9: Algorithme à Trous.

### 4.3.3 Schéma avancé de codage par ondelettes : les paquets d'ondelettes

Comme il en est fait mention dans le paragraphe 3.2, toute décomposition s'efforce de produire des sous-bandes dont le spectre est le plus plat possible. D'une façon générale, la décomposition en ondelettes opère une très bonne décorrélation, en particulier pour des images dont le spectre décroît de façon exponentielle ou uniforme. Cependant, la technique reste moins performante pour des



signaux contenant des pics d'amplitude spectrale en hautes fréquences. Les paquets d'ondelettes, élaborés par Coiffman et Meyer [CM89], compensent ce défaut inhérent aux ondelettes classiques.

Les paquets d'ondelettes, comme les ondelettes de base, sont formés par une combinaison linéaire de fonctions d'échelles dilatées et translatées. La différence réside dans le fait que la décomposition est cette fois appliquée sur l'une des deux sous-bandes d'un niveau de résolution donné : l'opération n'est plus réservée à la sous-bande d'approximation. Le schéma 3.10 permet de visualiser la différence d'analyse du signal entre ondelettes classiques et paquets d'ondelettes.

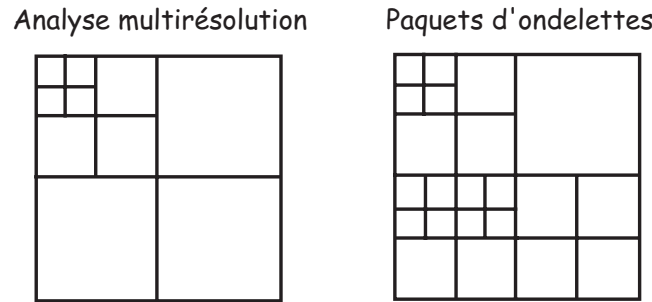


FIG. 3.10: Visualisation des différences de grilles issues d'une décomposition multirésolution et d'une décomposition en paquets d'ondelettes.

#### 4.4 Pourquoi la compression d'images par ondelettes est-elle efficace ?

De larges classes d'images peuvent être représentées de façon économique dans des bases d'ondelettes. En effet, les décompositions obtenues sont souvent dites "creuses" (peu de coefficients sont significativement différents de zéro [MMOP03]). Ces dernières années ont donc vu l'émergence de nombres de techniques de compression basées ondelettes. On peut citer à cet effet EZW [Sha93], CREW [ZASB95], SPIHT [SP96b], ECBOT [Tau99], ou encore la norme JPEG-2000 [BCM00].

## 5 Multirésolution et pyramide : État de l'art

Les analyses multirésolutions précédentes sont usuellement définies comme constituant une famille de sous-espaces emboîtés, possédant des propriétés bien particulières (voir paragraphe 2). Elles permettent une description du signal comme étant la somme orthogonale d'une approximation grossière et d'une infinité de détails plus fins. La décomposition en sous-bandes est considérée comme une décomposition pyramidale particulière.

Différentes statistiques peuvent être élaborées pour bâtir une pyramide d'images. La pyramide immédiate et la plus simple consiste en un sous-échantillonnage par 2 sur chacune des deux dimensions des images successives. Si le nombre de symboles récupérés est égal au nombre de pixels initial, cette décomposition ne peut toutefois pas constituer une représentation correcte de l'image. En effet, le défaut majeur introduit, à savoir l'*aliasing* (ou repliement de spectre), nuit considérablement. L'état de l'art fait donc mention d'autres types de construction de pyramide [Agh95].

Cette section recense donc les méthodes existantes, en distinguant pyramide à redondance de symboles, pyramide réduite, et décomposition en ondelettes. Une structure pyramidale faisant intervenir la notion de *niveaux*, par convention, nous affecterons le niveau 0 à la pleine résolution, et le niveau  $l_{max}$  au sommet de la pyramide (autrement dit l'image construite la plus petite).

### 5.1 Construction de la pyramide : méthodes à redondance de symboles

Les systèmes de codage pyramidal offrent un intérêt tout particulier. Burt et Adelson [BA83] ont été les tous premiers à se préoccuper de cet aspect de la compression d'images, en proposant la mise en œuvre d'une pyramide *gaussienne*, puis, ont fait évoluer cette notion par l'introduction de la pyramide *laplacienne*. Il est à noter, que si ces techniques conduisent à une reconstruction exacte, le nombre de symboles à transmettre s'avère plus important que pour les techniques basées ondelettes ou pyramides réduites (voir paragraphes suivants).

#### 5.1.1 Pyramide des moyennes

La pyramide des moyennes se calcule simplement : la moyenne issue de quatre pixels d'un niveau donné est la valeur d'un pixel du niveau supérieur de la pyramide. Cette méthode de construction pyramidale des données réduit les artefacts dus au sous-échantillonnage. Par ailleurs, cette représentation, au même titre que les pyramides gaussiennes, laplaciennes et résiduelles, définies dans les sous-sections suivantes, est redondante. Le nombre de symboles à transmettre est en effet supérieur au nombre de pixels de l'image originale. Pour un signal de longueur  $N_x \times N_y$  et  $J$  niveaux de décomposition, il est nécessaire de transmettre un nombre de symboles égal à

$$N_x \times N_y + \frac{N_x \times N_y}{4} + \frac{N_x \times N_y}{16} + \dots + \frac{N_x \times N_y}{(2J-1)^2} + \frac{N_x \times N_y}{(2J)^2} \xrightarrow{J \rightarrow \infty} \frac{4}{3} N_x \times N_y. \quad (3.9)$$

Puisque le calcul de la moyenne implique une troncature de la somme, cette méthode est aussi dénommée *Pyramide des moyennes tronquées*.

#### 5.1.2 Pyramide gaussienne

L'image originale, représentée initialement par  $w_0$ , constitue le bas de la pyramide, ou encore le niveau 0. Le niveau immédiatement supérieur, qui contient l'image  $w_1$ , est une version obtenue par filtrage passe-bas de  $w_0$ . Chaque valeur de  $w_1$  est calculée comme étant la moyenne pondérée des valeurs des pixels d'une fenêtre  $n \times n$ , au niveau 0. Les niveaux supérieurs sont obtenus après réitération du processus. La séquence d'images  $w_0, w_1, \dots$  est appelée pyramide gaussienne.

#### 5.1.3 Pyramide laplacienne

La pyramide gaussienne constitue la première étape de construction de la pyramide laplacienne. Les valeurs de  $w_1$  servent alors comme prédiction aux valeurs des pixels de l'image  $w_0$ . Pour parvenir à une représentation compressée de l'image, on encode l'image d'erreur  $E_0$  issue de la différence entre  $w_0$  et  $w_1$  interpolée notée  $\bar{w}_1$ . Cette image devient alors le bas de la pyramide laplacienne. La génération du niveau suivant se fait par l'encodage de  $w_1$  de façon similaire.

La pyramide laplacienne se définit alors comme étant la séquence d'image d'erreurs  $E_0, E_1, \dots, E_{l_{max}}$ . Chacune de ces imagettes est la différence de deux niveaux successifs  $l$  et  $l+1$  de la pyramide gaussienne. Ainsi, pour  $0 \leq l < l_{max}$ ,

$$E_l = w_l - EXPAND(w_{l+1}),$$

avec  $EXPAND$  formalisant le processus d'interpolation de l'image  $w_{l+1}$ ,  $\bar{w}_l = EXPAND(w_{l+1})$ , et  $E_{l_{max}} = w_{l_{max}}$  (il n'existe pas d'image servant à la prédiction de  $w_{l_{max}}$ ).

Au décodeur, l'image originale est reconstruite par la méthode suivante :  $\tilde{w}_{l_{max}}$ , image reconstruite au niveau  $N$  de la pyramide, est interpolée et additionnée à  $E_{l_{max}-1}$  pour former  $\tilde{w}_{l_{max}-1}$ . De même,  $\tilde{w}_{l_{max}-1}$  est interpolée et additionnée à  $E_{l_{max}-2}$ , et ainsi de suite, jusqu'à ce que le niveau 0 soit atteint.

La figure 3.11 illustre la construction d'une pyramide laplacienne à un niveau. Le signal passe par le filtre  $h_0$  passe-bas, puis est décimé afin de produire la sous-bande  $w_1$ . La sous bande  $E_0$  est formée par  $w_1$  sur-échantillonnée et convoluée avec le filtre d'interpolation  $EXPAND$ , et soustraite à l'image originale. Le signal est reconstruit au décodeur par le sur-échantillonnage et le filtrage par  $EXPAND$  de  $w_1$ , et additionnée à  $E_0$ .

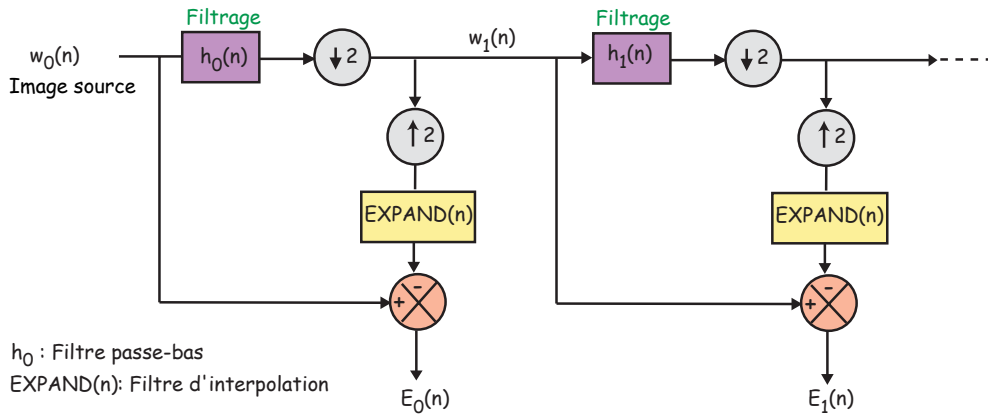


FIG. 3.11: Construction de la pyramide laplacienne pour un signal 1D

Dans ce schéma, la compression obtenue est due au fait que l'entropie, ainsi que la variance des images d'erreurs  $\{E_l\}_{l=0}^{l_{max}}$  sont réduites (histogramme très centré sur zéro). Enfin, Marr et Hildreth ont montré que le laplacien d'une gaussienne est en fait la limite d'une différence de deux gaussiennes : ceci explique le nom de la pyramide exprimée par Burt et Adelson.

#### 5.1.4 Pyramide d'images résiduelles

La pyramide laplacienne fait partie d'une famille de décompositions plus générales : les pyramides à prédiction/résidus. Les données disponibles sont utilisées pour prédire les suivantes. Cette prédiction se voit alors soustraite aux données suivantes afin de former une image dite résiduelle, qui est alors encodée. Le fait de ne pas utiliser une opération d'interpolation (comme la pyramide laplacienne) produit un ensemble de valeurs mieux décorréliées [CM95]. Les couches résiduelles alors construites ont une entropie et une variance plus faibles que celle des couches laplaciennes. C'est pourquoi ce type de pyramide s'avère efficace dans le domaine de la compression d'images.

## 5.2 Pyramides réduites

Les pyramides réduites s'inspirent largement de la méthode de la pyramide des moyennes. Toutefois, afin d'éviter toute redondance du nombre de symboles à transmettre, des solutions ont été trouvées. Dans la suite de cette étude,  $b$  correspond au nombre de bits nécessaires pour représenter un pixel de l'image à un niveau donné de la pyramide (typiquement  $b = 8$  pour des images pleine résolution en niveaux de gris).

### 5.2.1 Pyramide des sommes réduite

Pour cette pyramide, la somme des quatre pixels  $s = x_1 + x_2 + x_3 + x_4$  est conservée telle quelle (contrairement à la pyramide des moyennes simple, qui utilise la moyenne). Pour éviter tout dépassement,  $b + 2$  bits sont alloués pour le stockage de cette valeur. Pour le niveau inférieur, il nous suffit alors de ne prendre en compte que trois des quatre pixels  $\{x_i\}_{i=1\dots 4}$ . La redondance obtenue correspond à 8,3% de bits en plus de l'image originale.

### 5.2.2 Pyramide des différences réduite

Cette fois-ci, la moyenne des quatre pixels d'un bloc  $2 \times 2$  est stockée, ainsi que trois des valeurs présentées ci-dessous.

$$\begin{aligned} d_1 &= x_1 - x_2 \\ d_2 &= x_2 - x_3 \\ d_3 &= x_3 - x_4 \\ d_4 &= x_4 - x_1 \end{aligned}$$

Chacune de ces différences nécessite  $b + 1$  bits pour son encodage. Il en résulte 12.5% de bits en plus de l'image d'origine servant à l'encodage de la pyramide des différences réduite. L'avantage de cette technique réside dans une meilleure décorrélation des pixels à chaque niveau, et conduit donc à un encodage entropique plus efficace.

### 5.2.3 Pyramide à Transformée en S

La pyramide à transformée en S est une variante de la précédente technique [BF89]. La méthode requiert  $b$  bits à l'encodage de la moyenne. A cela, il faut ajouter le coût de codage des valeurs suivantes :

$$\begin{aligned} d_1 &= \frac{1}{2}(x_1 + x_2 - x_3 - x_4) & 9 \text{ bits} \\ d_2 &= \frac{1}{2}(x_1 - x_2 - x_3 + x_4) & 9 \text{ bits} \\ d_3 &= x_1 - x_2 + x_3 - x_4 & 10 \text{ bits} \end{aligned}$$

soit 16.7% de bits supplémentaires.

## 5.3 Pyramide et ondelettes

### 5.3.1 Décomposition classique

D'après les paragraphes précédents, la décomposition d'un signal par une pyramide laplacienne ou gaussienne possède le désavantage suivant : le nombre de symboles à transmettre dépasse celui

des valeurs initiales d'un facteur égal à  $4/3$ . A partir de la modélisation des pyramides ci-dessus, de nombreux travaux ont alors été entrepris afin de répondre à une propriété supplémentaire : l'orthogonalité de la transformation [WO86, ASH87]. Les bases de la décomposition pyramidale à transformée en ondelettes sont alors jetées (voir section 4). Les filtres passe-haut et passe-bas sont dans ce cas appliqués uniquement sur les coefficients d'approximation  $a_j(n)$  (fig. 3.12). Les coefficients de détails ( $h_j(n)$  : coefficients horizontaux,  $v_j(n)$  : coefficients verticaux,  $d_j(n)$  : coefficients diagonaux) de l'image sont directement transmis au décodeur. Pour tout niveau  $j$  de la pyramide, chaque matrice de coefficients est de taille  $4^j$  fois plus petite que l'image originale. La construction d'ondelettes orthogonales revient donc à la synthèse de filtres conjugués en miroir possédant une propriété de stabilité.

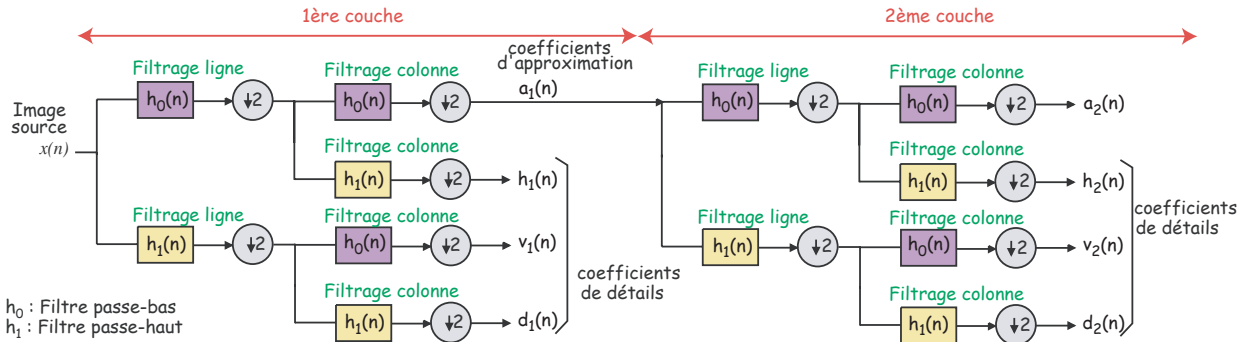


FIG. 3.12: Décomposition pyramidale en ondelettes pour un signal 2D par un codeur à deux couches.

Parmi l'ensemble des décompositions en ondelettes disponibles, la transformée en S retient notre attention. Décrite par Lux en 1977 [Lux77], elle a été fortement mise en avant par les techniques de représentation sans perte liées à l'imagerie médicale. En particulier, l'algorithme S+P, présenté au chapitre suivant, tire parti des propriétés de cette transformée et s'appuie sur une méthode prédictive afin d'améliorer le taux de compression d'une image donnée.

### 5.3.2 Ondelettes de seconde génération : le Lifting Scheme

La méthode du *lifting scheme*, due à Sweldens [Swe95], est une technique très flexible, qui permet de fabriquer une infinité de bases biorthogonales d'ondelettes en partant d'une base biorthogonale d'ondelettes donnée. Le lifting est utilisé dans de nombreux dispositifs de codage, afin de faciliter la construction et l'implantation des ondelettes traditionnelles [Swe96] et des ondelettes dites de seconde génération, comme les ondelettes sphériques [Swe98].

La figure 3.13 donne le schéma-bloc de la méthode du lifting (transformation directe), qui consiste à représenter un banc de filtres 1D sous la forme d'une structure en échelle. A chaque niveau de décomposition, une transformation triviale en ondelettes ("Lazy wavelet", dont le schéma bloc est délimité par un cadre en pointillé) est d'abord appliquée. Les échantillons d'indice pair sont séparés des échantillons d'indice impair. La structure en échelle fait alterner les pas suivants :

1. *Prédiction* ou *dual lifting* : les échantillons d'indice pair sont remplacés par la différence entre leur valeur et son estimation, calculée au moyen d'un filtre  $p_i(z)$ , en fonction des échantillons d'indices impairs du voisinage. Cette étape a un effet décorrélateur qui s'apparente fortement

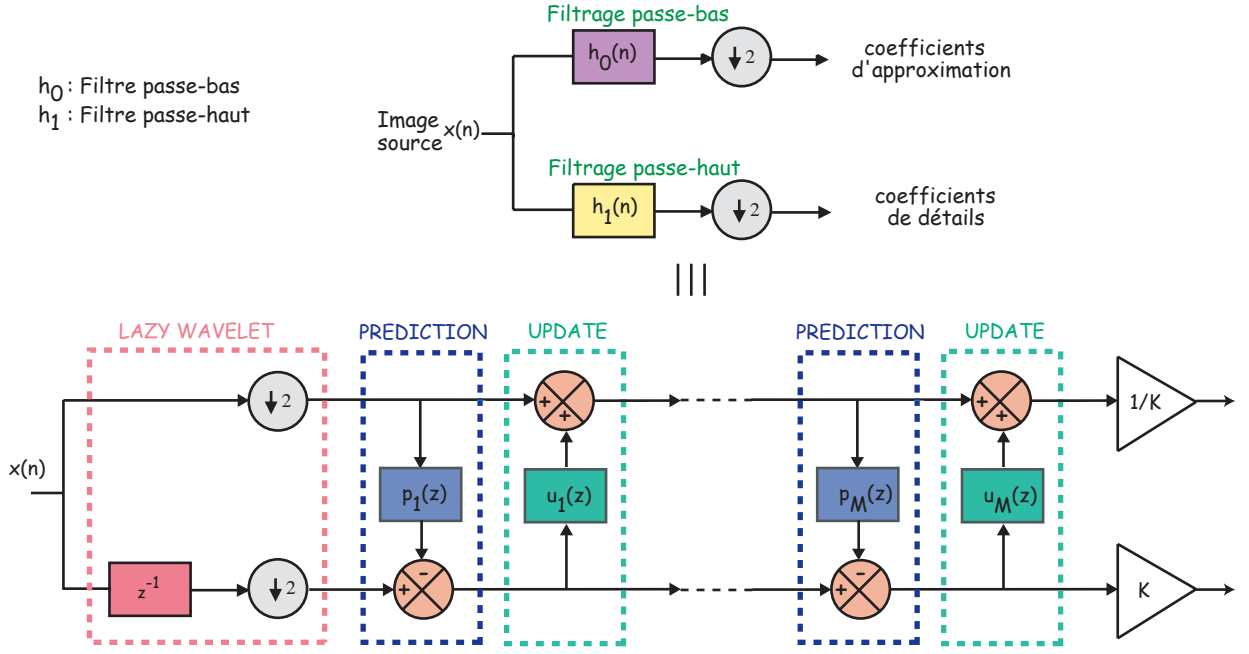


FIG. 3.13: Banc de filtre 1D et sa structure lifting équivalente

aux techniques de codages prédictifs du type MICD<sup>7</sup>, à la restriction suivante près : seuls les pixels situés sur la même ligne sont exploités. Enfin, ce procédé revient à appliquer un filtre passe-haut.

2. *Update* ou *lifting* : les nouveaux échantillons pairs permettent la remise à jour des échantillons impairs au moyen des filtres  $u_i(z)$ . Le deuxième pas de lifting s'apparente donc à un filtrage passe-bas appliqué sur les coefficients résiduels de l'étape de prédiction. L'effet dû au repliement de spectre induit par le sous-échantillonnage est ainsi corrigé.

Enfin, la dernière étape d'encodage est un pas dit de *scaling* qui applique un facteur correctif  $(K, 1/K)$ , à chacune des deux voies. Les valeurs corrigées correspondent alors aux coefficients d'approximation et de détail au niveau courant. Après  $M$  étapes de prédiction / mise à jour, les échantillons pairs constituent l'ensemble des coefficients d'approximation, et les échantillons impairs représentent les coefficients de détails.

Pour un signal unidimensionnel, l'algorithme 5.3.2 sert à l'obtention des coefficients d'approximation  $\{a_l(n)\}$ , et des coefficients de détails  $\{d_l(n)\}$  au niveau  $l$ , après  $n$  itérations.

<sup>7</sup>MICD : Modulation par Impulsions à Codage Différentiel

**Algorithme 3.3** Lifting scheme : algorithme général**Split :**

$$a_{l+1}(n) = a_l(2n)$$

$$d_{l+1}(n) = a_l(2n + 1)$$

**Predict / Update :****Pour**  $k = n \dots 1$ 

$$d_{l+1}(n) = d_{l+1}(n) - p_k(a_{l+1}(n)),$$

$$a_{l+1}(n) = a_{l+1}(n) + u_k(d_{l+1}(n)).$$

**fin Pour****fin Lifting**

Les avantages de la méthode sont les suivants [Swe96].

1. Le schéma tire parti des similarités existantes entre les filtres passe-haut et passe-bas, afin d'accélérer les calculs : l'implantation et l'exécution de la transformée en ondelettes est de ce fait plus rapide.
2. Aucune mémoire supplémentaire n'est nécessaire dans le schéma de codage : l'image originale est remplacée par les coefficients d'ondelettes.
3. La phase de reconstruction est obtenue très simplement, à l'aide des mêmes filtres que ceux utilisés par la méthode directe (fig. 3.14).

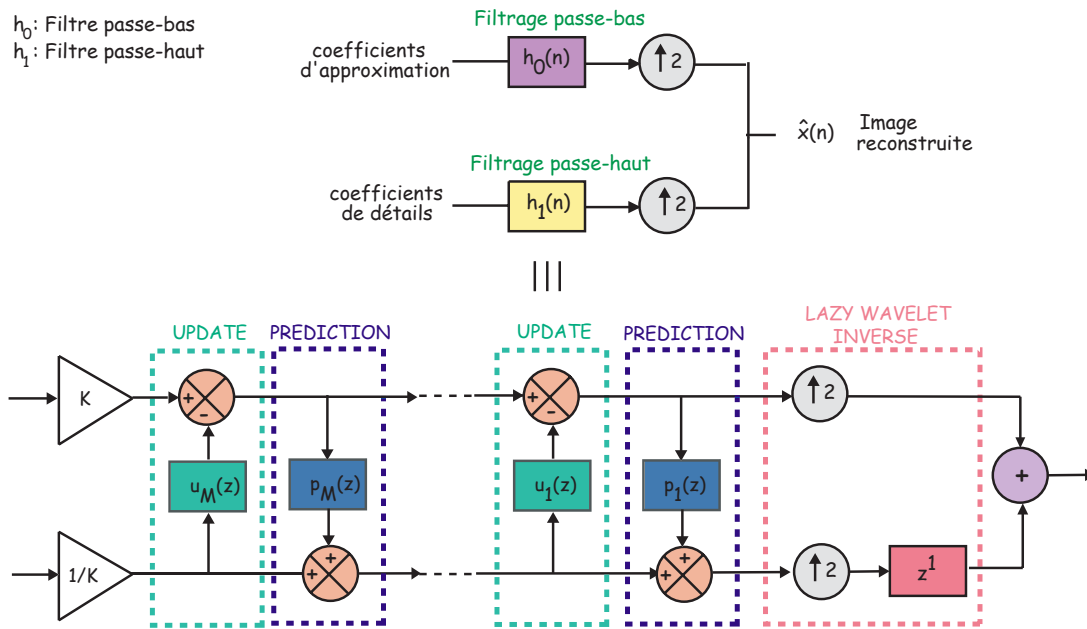


FIG. 3.14: Reconstruction en lifting d'un signal 1D

Nombre de travaux ont été entrepris sur des schémas à base de lifting, en particulier dans le cadre de la compression sans perte. Divers bancs de filtres à reconstruction parfaite ont donc été réalisés : la transformée à S+P [SP96a] et RTS (Reversible Two Six - utilisée par la méthode CREW [ZASB95]) en sont les exemples les plus aboutis. En définitive, le schéma de lifting peut produire une transformation d'entier en entier sans perte d'information, à condition de récupérer exactement les entrées entières du signal 1D. Une fois encore, le cas du signal 2D est traité par

application successive du lifting sur les lignes puis sur les colonnes.

## 5.4 Construction de la pyramide : les méthodes non linéaires

Un bon exposé des techniques à transformées multi-échelles non linéaires a été établi par J.-L. Starck [Sta02]. Nous présentons dans les sections suivantes les deux familles de méthodes non-linéaires les plus usitées, à savoir : les décompositions pyramidales morphologiques, ainsi que le codage fractal.

### 5.4.1 Décomposition pyramidale morphologique

L'idée est ici d'utiliser des filtres morphologiques pour la décomposition en sous-bandes, afin de remédier complètement aux problèmes des phénomènes de Gibbs, caractéristiques du sous-échantillonnage. Globalement, trois types de pyramides morphologiques coexistent :

1. les pyramides construites par itérations d'opérations morphologiques,
2. les pyramides basées sur l'itération d'un filtrage et d'un échantillonnage,
3. les méthodes "multirésolution" : une étape est ajoutée (extraction de caractéristiques puis décision).

Un état-de-l'art des pyramides morphologiques peut être trouvé dans [Lap02]. Nous nous contenterons d'évoquer ici les principales avancées effectuées dans ce domaine de recherche.

Dans un premier temps, les opérations d'ouverture et de fermeture ont été étudiées dans le cadre de la décomposition pyramidale d'un signal. Il a en effet été démontré que la morphologie mathématique s'apparente à l'utilisation de filtres passe-bas permettant d'aller d'une résolution à une autre résolution, approximation de la première (figure 3.15). Malheureusement, le théorème intitulé "Digital Morphological Sampling Theorem" annonce les limites de la méthode : les bancs de filtres ainsi conçus ne peuvent procurer ni un échantillonnage satisfaisant, ni une reconstruction parfaite [HZLL89].

C'est pourquoi, de nouvelles techniques [EVE97], mettant en œuvre des filtres médians à support de tailles et de formes variables, ont été élaborées. Du point de vue qualité visuelle, les images reconstruites sont agréables à l'œil humain : aucun effet de rebond n'est visible. Le désavantage majeur de cette technique est la pauvreté de la représentation de la texture à très bas débit.

Plusieurs solutions ont donc été proposées dans la littérature. Egger choisit de décomposer le signal en sous-bandes (sans redondance de symboles), en adaptant le choix des filtres en fonction du contexte [EVE97] (filtres morphologiques pour les zones homogènes, et filtres linéaires pour la texture). Starck, quant à lui, introduit les notions de transformée itérative, non-itérative, redondante ou non redondante, dans le cadre des constructions pyramidales à bancs de filtres médians [Sta02].

Les transformées multirésolutions basées médian présentent globalement l'avantage d'être robustes aux fortes singularités de l'image. Elles constituent un outil adapté en particulier à la compression des images astronomiques : les points particuliers correspondants aux étoiles peuvent alors être parfaitement séparés du reste du signal.



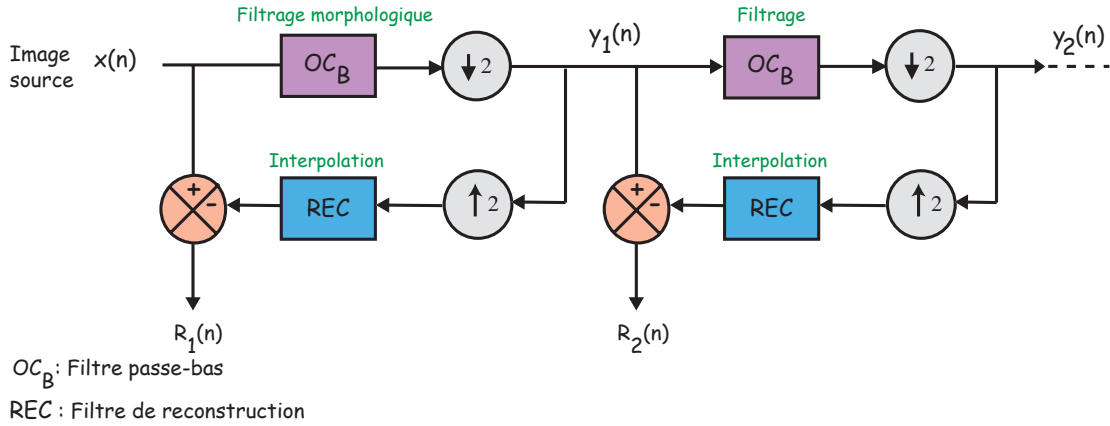


FIG. 3.15: Décomposition pyramidale morphologique. L'opération  $OC_B$  représente l'ouverture-fermeture par l'élément structurant  $B$ . La décomposition produit une image d'approximation  $y_L$  où  $L$  correspond au niveau le plus haut de la pyramide, et plusieurs images résiduelles  $\{R_i\}_{i=1}^{L-1}$ .

#### 5.4.2 Décomposition pyramidale à l'aide de fractales

Le propos n'est pas ici de traiter des fractales. Le lecteur se référera en particulier au rapport technique écrit par Chee [Che95] : un état-de-l'art des compressions fractales d'images y est établi. La compression par fractales est fondée sur la recherche d'autosimilarités dans l'image. Cette recherche est effectuée sur une représentation par blocs et l'autosimilarité consiste à approximer un bloc de l'image (appelé bloc destination) par un autre bloc de taille supérieure (appelé bloc source). L'approximation est réalisée selon une transformation affine contractante du bloc source sur le bloc destination. La propriété de contraction se situe aussi bien au niveau spatial qu'au niveau des intensités. L'union de telles transformations contractantes, appelée "système de fonctions itérées" (IFS en anglais) assure la convergence d'un processus itératif vers un point fixe, appelé attracteur, proche de l'image à coder. La transformation par fractales a été largement développée sur des partitionnements par blocs carrés.

Le codage fractal est donc basé sur la constatation suivante : les images naturelles possèdent des auto-similarités (répétition d'un motif) à différentes échelles. Un codeur fractal pyramidal a été proposé par Chee : chaque niveau de la pyramide est alors encodé au moyen d'une compression fractale. Cette méthode a de particulier la transmission progressive (voir §1.1) des données - fonctionnalité non naturelle des codages fractals. Un codage hybride fractale/sous-bandes a été élaboré [BDBC97], avec une génération de "Système de Fonctions Itératives Locales" (LIFS) adaptée et accélérée pour chaque sous-bande. Dans cette approche, si l'image est d'abord décomposée par des filtres orthogonaux, la recherche de l'auto-similarité est effectuée indépendamment dans chaque sous-bande. On cherche donc à exploiter la corrélation intra-sous-bande en générant un LIFS adapté à chaque sous-bande.

## 6 Conclusion

Ce chapitre a ainsi décrit les grandes étapes historiques de la construction de décompositions pyramidales d'une image fixe donnée, au profit d'une compression avec ou sans perte. Les éléments de base de la construction d'une analyse multirésolution ont donc été présentés, afin de mettre en lumière les enjeux de la décomposition en sous-bandes d'un signal donné. Les méthodes présentées ne peuvent être exhaustives : la multiplicité des schémas pyramidaux démontre cependant l'intérêt croissant porté à ce type d'encodage. En particulier, si nous avons simplement étudié les principes de base nécessaires à la compréhension du fonctionnement des ondelettes, de nombreux travaux récents portent sur des schémas avancés aux noms évocateurs : "arêtelettes" (ridgelets) ou encore ondelettes de seconde génération, et bandelettes, dernières nées issues des travaux conjoints de Le Pennec et Mallat [NM05].

La définition de la seule structure pyramidale ne peut constituer en soi une méthode de codage. Des stratégies de codage s'appuyant sur cette pyramide doivent donc être élaborées à des fins de compression progressive efficace. Le propos sera développé au chapitre suivant, et illustré par la description de codeurs issus de l'état de l'art.



## Chapitre 4

# Stratégies de codage efficace des décompositions pyramidales - Etat de l'art

Nous avons vu dans le chapitre précédent les différentes façons de construire des pyramides dans le cadre de la compression d'images. Nous allons présenter ici les méthodes de codage pyramidal efficaces existantes. Une fois la décomposition et la quantification éventuelle (pour de la compression avec pertes) effectuée, on dispose d'une suite d'éléments appartenant à un ensemble fini de symboles, que l'on cherche à coder le plus économiquement possible. En effet, la décomposition à elle seule ne constitue pas une compression en tant que telle, mais une transformation de l'information dans un autre espace de représentation.

La localisation à la fois spatiale et fréquentielle des images sous-bandes a été exploitée à travers essentiellement deux types de techniques. La première tente de profiter des structures de données efficaces, comme les *zerotrees*, les arbres hiérarchiques, ou le conditionnement inter-sous-bandes. L'idée est alors d'utiliser la corrélation qui existe entre les sous-bandes, et la corrélation intra-sous-bande. Une deuxième solution de codage adopte des techniques dites adaptatives : les schémas alors élaborés permettent, d'une image à l'autre ou encore localement dans une même image, de changer de transformée (ou décomposition linéaire), de système de quantification ou de codage entropique.

Grâce aux efforts portés sur les algorithmes de compression d'images, en particulier ceux basés ondelettes, on peut dresser une liste des points essentiels de la chaîne de codage pour la définition d'un schéma progressif efficace. Pour cela, nous nous appuyerons tout particulièrement sur les travaux réalisés par Shapiro [Sha93], à la base de nombreuses innovations dans le domaine de la compression d'images.

Le chapitre s'organise de la façon suivante. Tout d'abord, la section 1 pose le problème du codage *embedded* des coefficients transformés. Nous introduirons de façon plus spécifique la méthode des plans binaires (Bitplane coding) (§1.2) : elle permet l'approximation successive des valeurs traitées. Le paragraphe 1.3 présente la notion de codage des *significance map*, -cartes binaires indiquant la position des coefficients significatifs-, ainsi que le partitionnement d'ensembles. Les algorithmes EZW, SPIHT et SPECK, basés sur l'exploitation des arbres à zéro (*zerotrees*) serviront à étayer le propos. Malgré leur efficacité, ces arbres possèdent des défauts inhérents à leur structure (§1.4). Des alternatives ont été trouvées, en particulier via le codage conditionnel

et contextuel des symboles (section 2). Dans ce cadre, nous verrons la définition des codeurs CREW et EBCOT. Par la suite, et ce afin de recentrer l'étude sur nos propres préoccupations, nous nous intéresserons aux méthodes pyramidales, à prédiction dans l'espace transformé (§3.1) ou dans le domaine spatial (§3.2). Enfin, dans la section 4, le standard JPEG-2000, sur lequel nous nous appuierons pour des comparaisons référencées de qualité visuelle d'images produites.

## 1 Stratégies d'encodage progressif des valeurs

Cette section présente les techniques d'encodage *embedded* les plus usitées dans la littérature. Dans un premier temps, le *Run-Length Coding* est décrit, même s'il ne fait pas partie des méthodes purement progressives : cet algorithme est un outil de base, le plus souvent associé aux arbres. Il nous paraissait donc judicieux de l'exposer ici.

Le codage de plans binaires (*Bitplane Coding*) constitue une façon très simple d'obtenir un code *embedded* (paragraphe 1.2). Associé aux arbres à zéros (*zerotrees*, §1.3), il permet d'obtenir des flux totalement scalables, au sens des notions présentées au chapitre 3. Les inconvénients de ce type de traitements sont mis en lumière, afin d'introduire, dans les sections suivantes, des techniques évoluées d'encodage progressif.

### 1.1 Une technique préliminaire : l'encodage par plages - *Run-Length Coding*

L'encodage par plage (*Run-Length Coding*) [Jai81], est basé sur le fait qu'il est fréquent de rencontrer une suite de plusieurs pixels voisins ayant des valeurs identiques. Une telle séquence de valeurs est alors appelée plage, caractérisée par 3 paramètres : l'adresse du début de la plage (soit le premier pixel de la plage courante), sa longueur, ainsi que sa valeur. Si toute l'image ne supporte que deux valeurs, comme c'est le cas pour une image binaire, les valeurs ne sont spécifiées qu'une seule fois et, par la suite, les longueurs des plages sont déterminées alternativement. La compression d'une image multi-niveaux revient à représenter l'ensemble de ces trois paramètres pour toute l'image. Cette méthode est d'autant plus efficace que le nombre de niveaux de gris dans l'image est faible. Elle est réversible et est particulièrement adaptée aux images binaires.

### 1.2 Codage de plan binaire - *bitplane coding*

Dans le cadre des codes dits *embedded*, ordonner les coefficients transformés selon un critère de qualité, puis les transmettre, ne constitue pas une solution unique : chacun des bits représentant ces coefficients peuvent aussi être ordonnés. Tout d'abord, le signe et l'amplitude sont encodés séparément. Le bit de poids fort de l'amplitude de chacun de ces coefficients est d'abord envoyé, puis le bit de poids immédiatement inférieur, etc... jusqu'à ce que l'ensemble des valeurs soient transmises et approximées. Cette méthode revient à adopter la démarche suivante : chaque valeur à coder est successivement quantifiée en divisant par deux l'intervalle dans lequel elle est représentée afin d'envoyer un bit qui indique le sous-intervalle retenu (fig. 4.1).

En pratique, le codage de plans binaires s'effectue en deux passes sur la collection entière des coefficients à coder. Deux listes sont alors établies : la liste dominante (passe significative) et la liste



Nous en exposerons les schémas principaux.

### 1.3.1 Construction des zerotrees - L'algorithme EZW

Si Lewis et Knowles utilisent pour la première fois, une arborescence basée sur une pyramide multirésolution [LK92], l'algorithme EZW (Embedded Zerotrees of Wavelets) développe une méthode de raffinement progressif de l'image [Sha93]. Considérons un coefficient inclus dans une sous-bande donnée comme étant parent<sup>1</sup>. A l'exception de la sous-bande de plus basse fréquence, chaque parent admet quatre fils dans la sous-bande de même orientation et de résolution immédiatement supérieure (fig. 4.3).

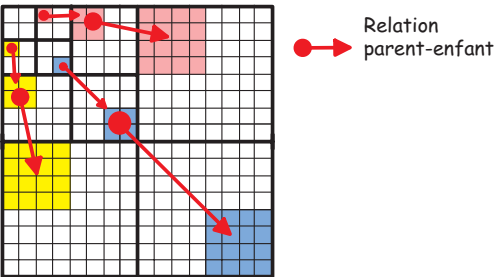


FIG. 4.3: Filiation entre les sous-bandes.

EZW se base sur la constatation suivante : si un coefficient dans une sous-bande donnée est nul, la probabilité pour que ses quatre fils soient également nuls est très forte. Ainsi, pour un coefficient est trouvé comme étant non significatif, sa descendance induite peut être écartée du processus de codage. Il est cependant possible qu'un ou plusieurs coefficients fils d'un parent non significatif soient néanmoins significatifs. Dans ce cas, un symbole codant l'information "zéro isolé" est transmis. La probabilité de cet événement étant faible, son coût de codage devient important. C'est le prix à payer pour éviter l'élimination de données significatives, dont les répercussions le long de l'arbre provoqueraient des distorsions considérables. Le *zerotree* est comparable à une forme évoluée et astucieuse de *run-length coding*, où les coefficients sont ordonnés afin d'engendrer de plus grandes longueurs de plage. De plus, les plages constituées possèdent leurs propres points d'arrêt : il n'est alors plus nécessaire de transmettre leur longueur. L'ensemble des coefficients significatifs est ensuite codé par application du codage de plan binaire suivie d'un codage entropique de type arithmétique adaptatif.

Cet algorithme, exploitant de façon intelligente l'information arborescente, n'est pas pour autant une méthode multirésolution au sens spatial, mais exprime plutôt la scalabilité en termes de qualité du flux.

### 1.3.2 SPIHT (Set Partitioning In Hierarchical Trees)

Said et Pearlman ont proposé une variante très efficace du schéma décrit par Shapiro. La description du codeur SPIHT [SP96b] propose la définition d'une nouvelle filiation entre coefficients. La différence principale réside dans le fait que, contrairement à EZW, le sens de parcours des arbres n'est pas spatialement figé, et ce afin d'optimiser le taux de distorsion. Ainsi, les arbres susceptibles

<sup>1</sup>La notion de bandes et de sous-bandes est explicitée au chapitre 3, §3

d'engendrer une importante réduction de la distorsion sont encodés en premier. Les meilleures performances affichées par SPIHT sur EZW sont également dues à une classification plus efficace des symboles source, ainsi qu'à l'utilisation d'une entropie conjointe lors du codage arithmétique. Il est à noter que, pour SPIHT, certains coefficients de la sous-bande basses fréquences ne possèdent pas de coefficients fils.

En résumé, SPIHT utilise des ensembles d'arbres spatialement orientés, dont les partitions des ensembles des coefficients fils et leur descendants sont générées par application d'un seuil sur l'amplitude de ces coefficients.

### 1.3.3 SPECK (*Set Partitioned Embedded Block*)

Contrairement à SPIHT, l'algorithme SPECK tente de décrire des partitions d'ensembles de coefficients à l'intérieur d'une même sous-bande [IP99]. Il s'apparente à un codeur basé blocs, qui toutefois emploierait un codage totalement progressif et *embedded*, par le codage de plans binaires ordonnés de façon décroissante. La structure hiérarchique de la pyramide est exploitée à travers une partition en octave, où l'énergie, concentrée dans les niveaux supérieurs, diminue graduellement lors de la descente de la pyramide. De plus, SPECK tire parti d'une décomposition en quadtree de l'image : en "zoomant" dans les zones de forte énergie d'une région donnée, celle-ci est alors codée par les techniques de *significance map* utilisées par EZW et SPIHT.

Ces caractéristiques font de SPECK un algorithme efficace, peu gourmand en ressources (vitesse et mémoire) et de faible complexité, au regard de son prédécesseur SPIHT et du standard JPEG 2000. Son atout principal réside dans le fait qu'il associe scalabilité en distorsion, et multirésolution spatiale.

## 1.4 Inconvénients des méthodes basées *zerotree*

Le développement du codeur EZW a suscité de nombreux travaux dans le domaine de la compression par *zerotree*. La simplicité intrinsèque de ces arbres, leur faible temps de calcul, ainsi que la possibilité de générer un code *embedded* sont des éléments évidemment attractifs. Cependant, si le codage par *zerotree* des symboles à transmettre est considéré comme très performant, il perd son efficacité lors de l'encodage des bits les moins significatifs. Si cet état de fait est tolérable dans la plupart des systèmes de codage avec perte, il n'en va pas de même pour la compression sans pertes. Par ailleurs, des travaux récents [MB01] ont démontré la capacité limitée des algorithmes *zerotree* à tirer parti des relations parents-enfants.

De plus, l'arborescence, telle celle imposée par les algorithmes du type EZW ou SPIHT, induit une structure artificielle sur les coefficients d'ondelettes. Seuls des contextes de forme carrée dans le domaine spatial sont utilisés, alors que statistiquement, les zones de coefficients connexes peuvent constituer des régions de forme arbitraire.

C'est pourquoi de nombreuses autres techniques, succinctement décrites dans la section ci-après, proposent un schéma de codage basé sur l'exploitation du contexte de chaque symbole à encoder.



## 2 Codage conditionnel et contextuel

L'ensemble des constatations, mentionnées dans le paragraphe précédent, ont conduit la communauté des traiteurs d'images à se pencher sur des solutions de compression non basées sur les *zerotrees*. Typiquement, les schémas de compression sans perte requièrent des techniques de codage entropique plus sophistiquées que pour une compression avec pertes : l'encodage conditionnel associé à une modélisation de contexte est devenu une composante indispensable des méthodes efficaces. Dans cette section, est tout d'abord introduite la définition d'un contexte, avant une présentation de quelques exemples d'encodage des sous bandes par observation du contexte.

### 2.1 Définition du contexte - Modélisation de contexte

Le contexte est défini comme étant un ensemble de données précédemment transmises et utilisées pour caractériser, par prédiction, la donnée courante. Le codage MICD ainsi que le codage entropique adaptatif sont des briques essentielles de nombreux exemples de prédictions basées contexte.

Wu, dans [Wu01], formule le problème de la modélisation statistique de contexte. Il fait la constatation suivante : puisque les transformées ne peuvent complètement décorrélérer le signal traité, des dépendances statistiques d'ordres supérieurs existent entre les coefficients transformés. De ce fait, une compression optimale peut être déterminée, si et seulement si il est possible de donner une modélisation de contexte d'ordre important. Wu décrit aussi les problèmes de **dilution de contexte** qui peuvent apparaître pour des ordres supérieurs : l'estimation des probabilités liées aux symboles à coder devient extrêmement complexe si le nombre de contextes est trop important. En conséquence, et paradoxalement, la longueur de code augmente lorsque l'ordre de la modélisation de contexte devient trop important ! Il convient donc de trouver des astuces nous permettant de réduire le nombre de contextes exploitables. On peut citer à cet effet la quantification de contexte (regroupement de contextes de même distribution statistique), ou encore la mise en place de certains discriminants.

### 2.2 Exploitation du contexte pour de la compression des images en sous-bandes

Dans ce paragraphe, seuls les schémas de codage intégrant la notion de contexte sans l'utilisation de techniques prédictives du type MICD sont mentionnés.

#### 2.2.1 *Zerotrees et information contextuelle - LZC*

Le premier schéma de codage de plans binaires s'affranchissant des contraintes inhérentes aux arbres semble être celui proposé par Taubman et Zakhor [TZ94] : Layered Zero Coding (LZC). Des quantificateurs, à nombre de bits décroissant, sont successivement appliqués à la structure en sous-bandes, et chaque *significance map* est encodée par un **codage arithmétique adaptatif**. LZC exploite le fait que le symbole le plus probable résultant de la quantification est le symbole zéro, ainsi que la dépendance spatiale intra-bande des données à coder. L'implantation de ce

codeur nécessite la définition de *code blocks* (partition rectangulaire de chaque sous-bande), à la base de EBCOT [Tau99], implanté dans le standard JPEG2000 [BCM00] (voir section 4). Après application de cette méthode, en comparaison avec les codeurs de Shapiro [Sha93] et de Said et Pearlman [SP96b], les auteurs ont constatés une nette amélioration du taux de compression.

### 2.2.2 Encodage conditionnel - Classification

Le contexte peut également servir de critère à la définition de plusieurs classes de coefficients pour chaque sous-bandes d'une image donnée. Le travail de Joshi et al. [JK<sup>+</sup>97] présente deux algorithmes (classification à gain maximum et classification à écart-type moyen indentique) : chaque bloc de coefficients dans une sous-bande donnée est attribué à une ou un nombre fini de classes définies comme étant des sources gaussiennes. La dépendance entre les classes de régions spatialement adjacentes est aussi prise en compte. Le but de la classification est ici d'affecter à toute classe un quantificateur choisi parmi un ensemble. La méthode de Joshi présente cependant un gain de classification moins important pour les très bas débits. L'article [YOY99] tente de compenser ce défaut par une classification contextuelle définie par observation des données précédemment quantifiées (adaptation arrière).

La méthode dite "Estimation Quantization" (EQ) modélise les sous-bandes comme des gaussiennes généralisées non stationnaires, l'énergie variant lentement dans chaque sous-bande. EQ suppose que le voisinage causal (donc déjà quantifié) d'une donnée (valeur parente incluse) possède la même énergie (variance) que la donnée elle-même. Ce codeur, plus efficace que SPIHT, n'est cependant pas complètement progressif.

Enfin, ECECOW [Wu97] (Embedded Conditional Entropy Coding of Wavelet Coefficients) est un codage entropique "embedded" d'ordre élevé et particulièrement efficace, qui s'applique indifféremment au codage avec pertes et sans perte.

## 3 Pyramides et méthodes prédictives

La technique dite "MICD" (Modulation par Impulsions Codées Différentielles) a été inventée par Cutler en 1952. Si les schémas simples de codage avec pertes par MICD ont été largement dépassés, en termes de performances, par les méthodes de compression par transformées, de nouvelles recherches ont émergé, combinant pyramide et prédiction. En particulier, les schémas efficaces mettent en place une décomposition pyramidale en boucle fermée (la représentation en boucle ouverte donnant de moins bons résultats [HWG97]), afin, dans le cadre d'une compression avec pertes, de prendre en compte l'erreur de quantification avant l'encodage des niveaux suivants.

Dans cette section, nous ferons la distinction entre les méthodes de prédiction opérées dans le domaine transformé, et celles réalisées dans le domaine spatial. L'étape finale de codage entropique n'est pas ici mentionnée : le lecteur pourra se référer aux techniques précédemment décrites.

### 3.1 Prédiction dans l'espace transformé

L'avènement des méthodes par transformées en ondelettes a fait porter la réflexion sur le codage inter- et intra-bandes des images des coefficients transformés. L'accent sera mis ici sur l'algorithme S+P (§3.1.1), avant de faire un bref tour d'horizon des méthodes alliant prédiction et transformation. Remarquons par ailleurs que l'utilisation de transformées réversibles permet de produire des méthodes de compression sans perte progressives.

#### 3.1.1 Un précurseur : l'algorithme S+P

La transformée en S, par l'application successive de la transformée Hadamard  $2 \times 2$ , construit une pyramide réduite très usitée dans le domaine de l'imagerie médicale. La méthode proposée par Said et Pearlman [SP93][SP96a] s'avère être la référence en matière de compression multirésolution sans perte. Elle utilise un schéma à décomposition pyramidale amélioré par un codage prédictif : la transformée S+P (transformée en S et prédiction). L'originalité réside dans le fait que l'étape de prédiction est effectuée *durant* les opérations successives de transformation.

*La transformée en S.*

La transformée en S fait partie de la famille des ondelettes discrètes (voir chapitre 3, §4). Si Lux[Lux77] fut l'inventeur de cette décomposition, elle a été grandement utilisée et réécrite afin d'améliorer ses performances calculatoires.

Soit  $a_l(n)$  une séquence d'entiers de longueur  $N$  au niveau  $l$  de la pyramide. Cette suite de symboles peut être représentée par deux séquences de longueur moitié définies par

$$\begin{aligned} a_{l+1}(n) &= \lfloor (a_l(2n) + a_l(2n+1))/2 \rfloor, \\ d_{l+1}(n) &= a_l(2n+1) - a_l(2n), \end{aligned} \quad (4.1)$$

où  $\lfloor \cdot \rfloor$  correspond à l'arrondi à l'entier inférieur. Les séquences  $a_{l+1}(n)$  et  $d_{l+1}(n)$  forment la transformée en S de  $a_l(n)$ , très semblable à la représentation multirésolution de Haar. La transformée bidimensionnelle est réalisée par l'application successive de cette transformée sur les lignes et sur les colonnes de l'image. La transformée en S est inversible selon

$$\begin{aligned} a_l(2n) &= a_{l+1}(n) - \lfloor d_{l+1}(n)/2 \rfloor, \\ a_l(2n+1) &= a_{l+1}(n) + \lfloor (d_{l+1}(n) + 1)/2 \rfloor. \end{aligned} \quad (4.2)$$

A cette écriture classique de la transformée en S, on peut ajouter la forme généralisée sous forme de lifting [CDSY98] (voir définition du lifting au chapitre 3) :

$$\begin{aligned} d_{l+1}(n) &= a_l(2n+1) - a_l(2n), \\ a_{l+1}(n) &= a_l(2n) + \lfloor d_{l+1}(n)/2 \rfloor, \end{aligned} \quad (4.3)$$

Même si maintenant la transformée est non-linéaire, le lifting permet de retrouver immédiatement son inverse :

$$\begin{aligned} a_l(2n) &= a_{l+1}(n) - \lfloor d_{l+1}/2 \rfloor, \\ a_l(2n+1) &= d_{l+1} + a_l(2n) \end{aligned} \quad (4.4)$$

La transformée  $S+P$ .

La transformée  $S+P$  exploite, lors de chaque transformation unidimensionnelle à un niveau  $l$  donné, certains échantillons des séquences  $a_l$  et  $d_l$ , afin d'estimer la valeur d'un  $d_l(n)$  donné. Le principe est le suivant : si  $\tilde{d}_l(n)$  est cette estimée, la différence

$$d'_l(n) = d_l(n) - \lfloor \tilde{d}_l(n) \rfloor$$

remplace  $d_l(n)$  pour former une nouvelle image transformée d'entropie de premier ordre plus faible. De nouveau, les signaux à deux dimensions sont traités par l'application de  $S+P$  sur les lignes et sur les colonnes.

La transformée s'écrit donc sous la forme

$$\begin{aligned} \tilde{d}_{l+1}(n) &= a_l(2n+1) - a_l(2n) \\ a_{l+1}(n) &= a_l(2n) + \lfloor \tilde{d}_{l+1}(n)/2 \rfloor \\ d_{l+1} &= \tilde{d}_{l+1}(n) + \\ &\quad \lfloor \alpha_{-1}(a_{l+1}(n-2) - a_{l+1}(n-1)) + \alpha_0(a_{l+1}(n-1) - a_{l+1}(n)) + \\ &\quad \alpha_1(a_{l+1}(n) - a_{l+1}(n+1)) - \beta_1 \tilde{d}_{l+1}(n+1) \rfloor \end{aligned} \quad (4.5)$$

Parmi l'ensemble des valeurs de coefficients possibles, Said et Pearlman ont pu empiriquement déterminer, dans le cas des images naturelles, le quadruplet optimal, à savoir :  $\alpha_{-1} = 0$ ,  $\alpha_0 = 2/8$ ,  $\alpha_1 = 3/8$ , et  $\beta_1 = -2/8$ . Notons par ailleurs que la transformée TS (Two-Six), notamment utilisée par l'algorithme CREW, est un cas particulier de cette transformée  $S+P$  : il suffit de choisir lors de la reconstruction les valeurs suivantes :  $\alpha_{-1} = \beta_1 = 0$  et  $\alpha_0 = \alpha_1 = 1/4$ .

Il est à noter que les auteurs de ce travail ont aussi décrit une adaptation de  $S+P$  à l'algorithme SPIHT (voir §1.3.2) : la transmission du flux est donc totalement progressive (au sens "embedded" du terme).

### 3.1.2 CREW : une compression efficace ?

CREW (Compression via Reversible Embedded Wavelet) [ZASB95] utilise un schéma similaire à celui présenté par Shapiro, conjointement à une méthode dite "Horizon". Celle-ci profite de l'information spatiale et fréquentielle présente dans le domaine des ondelettes.

Cet algorithme fournit un schéma de codage avec ou sans perte (un seul codeur). La progressivité par niveaux de résolution est obtenue par la décomposition par ondelettes entières, assurant ainsi le caractère réversible de la méthode. A cet effet, les transformées en  $S$  et  $TS$  sont utilisées alternativement (figure 4.4) [SZB96].

Le caractère "embedded" du code produit est le plus souvent lié à la quantification successive d'un symbole au moyen d'un encodage par plans binaires : la scalabilité par niveaux de qualité est ainsi réalisée. Associée à une modélisation de contexte et un codeur entropique ("Q-coder"), la technique assure une compression efficace, pyramidale et progressive.

Par ailleurs, la décomposition est idempotente. Ceci signifie qu'une image donnée, décompressée après codage avec pertes, peut être à nouveau recompressée avec pertes : le flux sera exactement égal au flux de la première image encodée, et la nouvelle reconstruction identique à la première. Enfin, CREW est relativement simple à développer aussi bien pour du

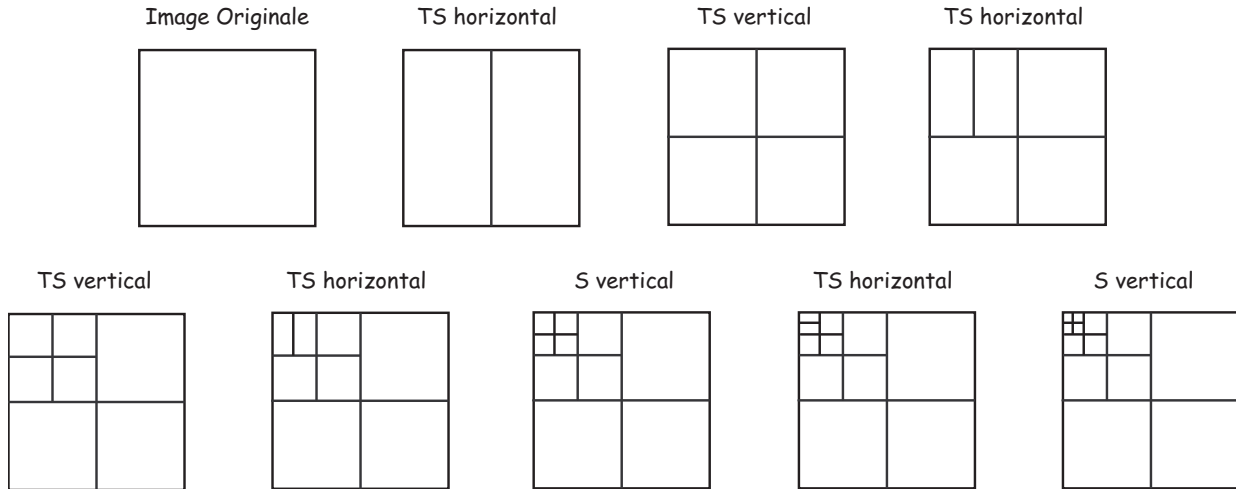


FIG. 4.4: Décomposition en ondelettes de CREW : alternance de la transformée en S et la transformée TS.

logiciel que pour une implantation matérielle.

Néanmoins, Said et Pearlman ont noté dans leur article [SP96c] d'étranges similarités entre CREW et leur propre codeur S+P, proposé un an auparavant. Ils ont aussi démontré que la proposition d'utiliser la transformée TS n'est qu'un cas particulier de leur définition de la transformée en S (voir paragraphe précédent).

### 3.1.3 Codage pyramidal prédictif et espaces transformés : un aperçu

De nombreux travaux ont été effectués sur la façon de réduire l'entropie de premier ordre des coefficients transformés. Nous pouvons cependant essayer de résumer les différentes méthodes actuellement proposées.

Un élément peut être prédit soit à partir de ses voisins, soit à partir des valeurs des pixels de l'image basse résolution (si cet élément appartient à une résolution supérieure), ou à partir de la combinaison des deux. Une solution pour la recherche du prédicteur optimal est de résoudre les équations de prédiction de Yule-Walker.

Par exemple, après application de la transformée de Haar modifiée [PM96], Piscaglia résout le système d'équations cité auparavant et obtient un jeu de prédicteurs linéaires, augmentant considérablement le taux de compression. Des tentatives visant à surpasser les performances de codage progressif de SPIHT se sont largement inspirées des techniques de prédiction fondées sur l'observation du contexte (par ex. [CO96]). L'article [ICB<sup>+</sup>00] construit un schéma sur une transformée en ondelettes sans perte associée à une prédiction de type MICD sans perte multiéchelle.

L'étude de Fukuma et al. [FIBK99] présente quant à elle un schéma de codage prédictif sur les coefficients d'ondelettes issus des 3 composantes couleurs (prédiction inter- et intra-sous-bandes, prédiction inter-composantes).

Dans le domaine des fractales, des études récentes ont montré que les fonctions contractantes, explorant la redondance entre les différentes résolutions de l'image, peuvent être vues comme une prédiction inter-échelle pour les coefficients d'ondelettes. C'est pourquoi des codeurs mettant en

œuvre des prédictions fractales, conjointement à une transformée en ondelettes, ont été élaborés (par ex. [LK99, BBHH97]). Toutefois, il s'avère le fait que ce type de schéma ne donne pas de résultats satisfaisants en termes de qualité visuelle.

### 3.2 Prédiction dans le domaine spatial

La mise en place d'un système de compression pyramidal dans le domaine spatial engendre un surcoût en termes de symboles (chapitre 3 §5.1). Dans ce contexte, un gain significatif peut être apporté par l'élaboration de schémas prédictifs, exploitant au mieux, pour un pixel donné, l'information déjà reconstruite.

La première méthode de codage hiérarchique efficace à mettre en œuvre une technique de prédiction est nommé HINT (Hierarchical INterpolation) [EY86]. L'image est simplement sous-échantillonnée en retenant un pixel sur 16. Cette imagerie "basse résolution" est traitée par un MICD simple. Les autres pixels sont reconstruits par interpolation : l'erreur commise est de même transmise et additionnée à la valeur interpolée pour la reconstruction de l'image. La figure 4.5 illustre l'ordre dans lequel les pixels sont traités : la première passe, visualisée sur le schéma, concerne bien évidemment l'image basse résolution précédemment décrite. Les interpolations successives sont savamment agencées suivant des critères de distances entre les pixels.

De nombreuses versions dérivées de l'algorithme original HINT ont été proposées, et diffèrent seulement les unes des autres par la technique d'interpolation utilisée : RLP (Reduced Laplacian Pyramid) [AAB96], MEP (Minimal Entropy Pyramid) [HV95], IHINT (Interleaved Hierarchical INterpolation) [AFB97].

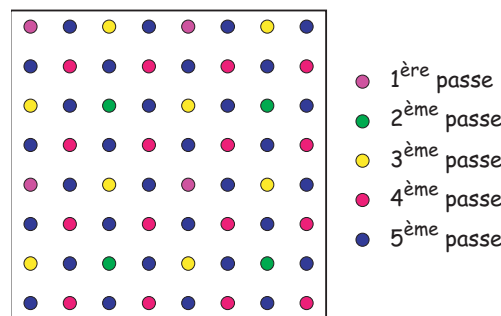


FIG. 4.5: Ordre d'encodage des pixels par la méthode HINT.

En 1992, Howard et Vitter mettent en place une méthode de modélisation de l'erreur commise par estimation de la valeur d'un pixel, et applicable à un schéma de compression hiérarchique sans perte [HV92]. Cette méthode prend appui sur le concept de *variability index* et fournit, pour les erreurs de prédiction, une modélisation précise des pixels permettant un gain de compression important. Les auteurs ont par la suite fait évoluer leur codeur par l'implantation du prédicteur FELICS [HV94]. Le modèle choisi est alors basé sur l'observation de quatre pixels voisins. Deux de ces voisins sont sélectionnés (valeurs médianes), puis, à partir de ces pixels, l'erreur entre valeur estimée et la valeur réelle est encodée. Le paramètre de codage est donc ajusté en fonction du contexte, et ce pour chaque niveau de la pyramide.

Rinaldo and Calvagno [RC95] ont proposé un codeur pyramidal prédictif (PPC<sup>2</sup>) par l'exploitation de la redondance inter-échelle. PPC accomplit une prédiction basée bloc par l'estimation des coefficients d'une bande donnée au moyen des coefficients des niveaux supérieurs de la pyramide. La prédiction utilisée est alors indépendante à chaque niveau de résolution, et de plus la taille des blocs peut être ajustée.

L'association de l'analyse multirésolution et des transformées polynomiales a été éprouvée par Escalante-Ramirez [ERVMGU95, VCER97]. La transformée polynomiale est en fait un modèle de représentation de l'image qui imite certaines propriétés du système visuel. Elle est ici utilisée pour modéliser les paramètres caractéristiques des contours présents dans l'image. Ainsi, les auteurs des articles précédemment cités parviennent à décrire une décomposition pyramidale prédictive, dont les symboles encodés correspondent respectivement à la moyenne locale, l'orientation du contour, la position du contour ainsi que son amplitude.

Des travaux récents ont été entrepris par Guangjin et al. [GFF97]. Les auteurs proposent l'utilisation d'un algorithme pyramidal à prédiction linéaire (PLPC<sup>3</sup>), dont l'efficacité n'est pas réellement prouvée (manque de résultats comparatifs).

## 4 JPEG2000 ou le développement d'un standard

JPEG-2000 est issu d'un projet initié par la construction du codeur CREW, précédemment décrit [BCM00]. Si JPEG-2000 utilise LZC et EZW, il intègre aussi le codeur EBCOT [Tau99], présenté en 1998 par Taubman : chaque sous-bande est divisée en blocs, encodés par *bitplanes* de façon indépendante les uns des autres. Un accès aléatoire aux informations spatiales est ainsi rendu possible. Il s'agit donc d'un standard pour lequel bien évidemment de nombreuses documentations sont disponibles : le lecteur pourra se référer à [SCE01b, SCE01a] par exemple. Nous ne présenterons donc ici que les grandes lignes de la méthode.

### 4.1 La norme de compression

Le projet JPEG-2000 a débuté en 1996, dirigé par Martin Boliek, et visait la création d'une méthode de compression innovante, intégrant notamment les fonctionnalités et les objectifs suivants :

- performances visuelles supérieures à l'état-de-l'art alors en vigueur,
- transmission scalable par qualité, résolution, composante ou région spatiale,
- compression avec ou sans perte,
- accès aléatoire au flux de données,
- fonctionnalités liées au zoom et au panorama (décompression d'une partie seulement du flux compressé),
- traitement dans le domaine compressé (par ex. rotation, recadrage...),
- encodage progressif des Régions d'Intérêt,
- implantation limitée en mémoire.

---

<sup>2</sup>PPC : Pyramidal Predictive Coder

<sup>3</sup>PLPC : Pyramid Linear Prediction Coding

Le standard est divisé en parties correspondant aux évolutions successives répondant à une fonctionnalité donnée<sup>4</sup>. La partie 1, cœur du système de codage, est désormais publiée et normalisée en temps que Standard International. Les parties 2 à 6 sont complètes ou en passe d'être finalisées, et concernent les éléments suivants : Extensions, Motion JPEG 2000, Conformance, Logiciel de Référence, Format d'images composées. Les parties 8 à 11 sont en cours de développement.

## 4.2 Architecture du codeur JPEG 2000

Dans ce qui suit, une description simple du système de codage est développée.

### 4.2.1 Prétraitement

Pour le standard, l'image peut être vue comme un ensemble de composantes (par exemple RGB), chacune susceptible d'être sous-échantillonnée par un facteur différent. L'image subit une étape préalable de division en un quadrillage régulier, dont le pas est modulable. Chacune des divisions produites s'appelle des **tuiles** ("tiles" en anglais). Chaque tuile d'une composante doit être de même taille (excepté le cas des bords de l'image). Le but de cette décomposition est d'une part de réduire la complexité de l'algorithme, et d'autre part de faciliter la navigation à l'intérieur des images traitées.

L'étape suivante consiste à retirer la valeur moyenne de chaque composante avant d'effectuer toute transformation (centrage de la dynamique). De plus, en présence d'une image couleur sous le format RGB, il convient de décorréler au préalable l'information par l'application d'une transformée (Transformée YCrCb ou Reversible Component Transform).

### 4.2.2 Transformées, quantification et encodage par blocs

Sur chacune des tuiles prétraitées, une décomposition en ondelettes est obtenue par application de l'algorithme de Mallat précédemment décrit. La DWT peut être réversible ou non. La transformée irréversible implantée par défaut met en œuvre les filtres de Daubechies 9/7 (voir signification chapitre 3, §4.3.1). Le filtre 5/3 de Le Gall est quant à lui utilisé dans le cas d'une transformation sans perte. Le standard autorise deux modes d'implantation de ces filtres : convolution classique ou lifting. La figure 4.6 illustre les étapes de traitement décrites jusqu'à présent.

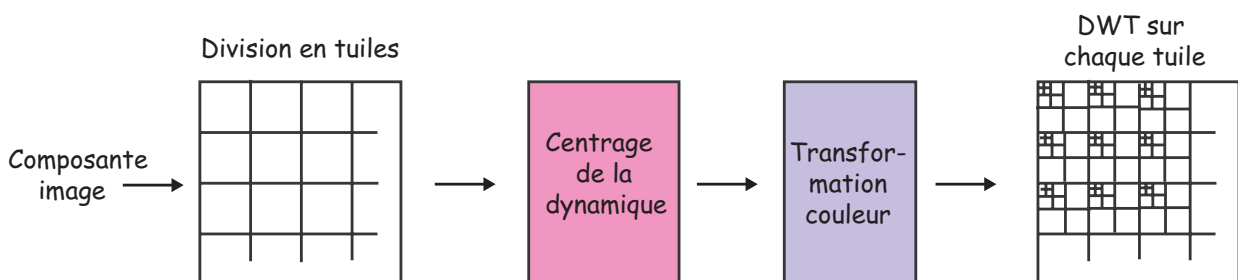


FIG. 4.6: JPEG 2000 : Prétraitement et transformée.

<sup>4</sup>Le lecteur pourra retrouver ces informations sur le site officiel de JPEG 2000 à l'adresse <http://www.jpeg.org/jpeg2000/>



L'algorithme JPEG 2000 intègre alors une phase de quantification linéaire sur chacune des sous-bandes produites : un pas de quantification est choisi par sous-bande construite. En règle générale, la quantification adoptée est fine pour les sous-bandes basses-fréquences (information visuelle importante), et grossière pour les hautes-fréquences.

Par la suite, chaque sous-bande issue des tuiles de base peut à son tour être divisée en "packet partitions" ou "**precincts**". Ces precincts correspondent à un regroupement de trois rectangles, de même localisation spatiale, mais appartenant aux trois sous-bandes de détails d'un niveau de résolution donnée. Ils seront partagés à leur tour en blocs rectangulaires plus petits ("**code-blocks**"). Ces notions sont représentées sur la figure 4.7.

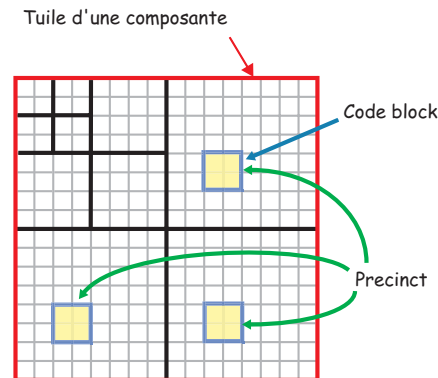


FIG. 4.7: JPEG 2000 : "Code-Block" et "Precinct".

Enfin, un codage entropique est exécuté de façon indépendante sur chacun de ces code-blocks. Ce processus est réalisé au moyen d'un encodage par plans binaires associé à un codeur arithmétique du type MQ, le tout formant un puissant codeur arithmétique adaptatif.

### 4.3 Regroupement par paquets et codage par couches

Pour chaque code-block d'un "precinct" donné, les flux compressés issus d'un certain nombre de plans binaires sont regroupés afin de former un **paquet**. Un paquet peut ainsi être interprété comme un apport de qualité pour un niveau de résolution, et ce pour une région spatiale déterminée. Un "**layer**", constitué d'un ensemble de paquets, peut permettre d'augmenter la qualité de l'image entière.

### 4.4 Performances de l'algorithme

Dues aux nombreuses innovations incluses dans l'algorithme, le standard JPEG 2000 présente des résultats concluant en termes de qualité d'image à bas débits. Une large part de la communauté des traiteurs d'images porte encore actuellement son effort sur l'amélioration de la technique, afin de garantir une haute qualité visuelle subjective. De plus, les diverses fonctionnalités proposées (compression avec pertes efficace, flux progressif, codage par régions d'intérêt, accès aléatoire, résistance aux erreurs...) en font un codeur de référence. Par contre, le développement de la partie sans perte de JPEG 2000 ne constitue pas l'état-de-l'art en la matière : l'introduction de la transformée en ondelettes 5/3 ne fournit pas une décomposition satisfaisante en termes d'entropie.

## 5 Remarques générales - Conclusion

L'état de l'art des codages pyramidaux permet de dégager les points essentiels à l'élaboration d'un schéma progressif efficace. La plupart des codeurs utilisent le codage par plans binaires, autorisant une transmission totalement scalable (progressivité à la fois au niveau résolution et qualité de l'image) des données. Associés à une exploitation de l'arbre de dépendance des données naturellement construit grâce à la structure pyramidale, ces codeurs (comme EZW, SPIHT, JPEG2000) affichent des performances intéressantes. Par ailleurs, comme largement démontré dans la littérature, une modélisation de contexte efficace se révèle d'une grande importance. Enfin, différents schémas prédictifs ont été proposés, réduisant l'entropie des symboles transmis. Il est à noter que les compressions les plus efficaces correspondent aux traitements dans l'espace transformé.

Nous introduisons dans les chapitres suivants trois méthodes de compression (LAR-APP, Interleaved S+P et RWHT+P) exploitant à la fois une décomposition pyramidale et les propriétés intrinsèques du LAR. Le travail réalisé porte sur une représentation et un codage pyramidal original, conduisant en particulier à l'élaboration de solutions alternatives aux techniques classiques de balayage des coefficients.



## Chapitre 5

# Approche pyramidale prédictive du LAR : Le LAR-APP

### 1 Introduction et objectifs

La construction d'un schéma de compression d'images par approche pyramidale prédictive vise à répondre à deux attentes fondamentales. La première consiste à proposer une méthode nouvelle scalable par niveaux en résolution et en qualité. Le second objectif est de proposer à l'utilisateur *une solution simple et unifiée de compression avec et/ou sans perte*. L'approche pyramidale basée sur la méthode LAR en est une réponse : un seul algorithme est utilisé à des fins de codage réversible ou non.

La plupart des schémas de codage par exploitation d'une pyramide associant une méthode prédictive travaillent dans un domaine transformé (voir chapitre précédent) et se trouvent être les plus efficaces. Nous présentons ici une décomposition pyramidale opérant dans le domaine spatial dont les différents niveaux sont traités par MICD dit "enrichi" en plusieurs passes. Outre l'exploitation d'un contexte à 360°, l'originalité réside dans le fait que cette pyramide est à redondance réduite, redondance éliminée systématiquement dans un cas de compression avec pertes. Les résultats - encourageants de notre point de vue - que nous avons obtenus ont fait l'objet de plusieurs publications. Le lecteur pourra donc se référer aux articles [BD03, BDR03c, BDR03a, BDR03d, BDR03b].

Ces travaux sont basés sur l'utilisation d'un prédicteur décrit par Wu en 1996 et présentés en section 2. Nous donnons le principe général de notre algorithme pyramidal dénommé LAR-APP<sup>1</sup> en section 3. Les sections 4 et 5 exposent respectivement les différentes étapes de construction et de descente de la pyramide. Nous relevons en particulier la propriété de redondance réduite de la méthode (§6). La section suivante 7 introduit alors la notion de *modélisation de contexte implicite* utilisée afin de réduire l'entropie des symboles codés. Par ailleurs, dans le cas de la compression sans perte, il est intéressant de savoir si le choix du seuil appliqué au gradient morphologique pour la construction du pavage influe sur le taux de compression (§8). Enfin, après avoir évalué l'efficacité du codage sans perte par comparaison avec l'état de l'art (§9), nous nous intéresserons à l'implantation de post-traitements sur nos images reconstruites avec pertes.

---

<sup>1</sup>LAR-APP : Approche Pyramidale Prédictive du LAR

## 2 Prédiction par l'algorithme de Wu

Wu proposa en 1996 [Wu97] un schéma de prédiction destiné à la compression sans perte des images et réalisant l'encodage de l'ensemble des données en trois passes entrelacées successives. Dans cette section, nous nous attachons à décrire son principe général par la présentation des différentes étapes de la méthode.

### 2.1 Motivations.

Comme la plupart des schémas MICD travaillent dans un ordre de balayage *raster*<sup>2</sup>, le contexte disponible est limité aux voisins déjà traités du pixel courant, éléments de l'espace causal. Ce type de configuration spatiale constitue un contexte de prédiction à 180°. Il est évident qu'il serait préférable de disposer d'un contexte dit à 360° entourant entièrement le pixel à coder : notre algorithme s'appuie donc sur cette idée de base. Nous avons vu, au chapitre précédent, des solutions de codage prédictif à 360° : on peut citer les méthodes HINT [EY86], MLP [HV92]... Cependant, les pixels utiles à l'estimation de la valeur courante sont distants du site à modéliser. Cet inconvénient empêche l'obtention d'un modèle réellement significatif utilisable dans un schéma MICD. Wu propose donc, dans le cadre d'une compression sans perte, de remédier à ce désavantage afin de fournir un contexte adjacent et enrichi pour l'ensemble des pixels à coder. Sa méthode se décline en trois passes successives décrites ci-après.

### 2.2 Sous-échantillonnage de l'image : application de la première passe.

Afin de construire ce contexte "amélioré", Wu sous-échantillonne tout d'abord d'un facteur 2 l'image originale  $I$  de taille  $N_x \times N_y$ . La première passe encode donc l'image  $\mu$  sous échantillonnée de taille  $N_x/2 \times N_y/2$ , composée de la moyenne de deux pixels diagonalement adjacents des blocs  $2 \times 2$  de l'image originale (Figure 5.1).

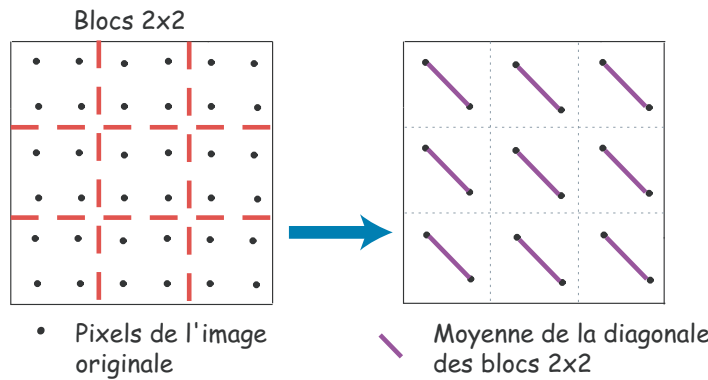


FIG. 5.1: Construction des moyennes de diagonales à coder.

Les valeurs des pixels de  $\mu$  s'expriment par la relation suivante :

$$\mu(x, y) = \left\lfloor \frac{I(2x, 2y) + I(2x + 1, 2y + 1)}{2} \right\rfloor, \quad 0 \leq x \leq N_x/2, 0 \leq y \leq N_y/2. \quad (5.1)$$

<sup>2</sup>raster scan : balayage des pixels ligne par ligne de l'image

Cette image est traitée à travers un MICD simple et classique opérant sur un contexte à  $180^\circ$  (Figure 5.2).

On note  $\check{\mu}$  la valeur prédite de  $\mu$ , décrite par la relation

$$\check{\mu}(x, y) = \frac{\mu(x, y-1) + \mu(x-1, y)}{2} + \frac{\mu(x-1, y-1) - \mu(x+1, y-1)}{4}. \quad (5.2)$$

L'erreur commise sur l'estimation de la valeur de  $\mu$  constitue la première information à encoder. On note ici l'utilisation d'un contexte de prédiction très simple.

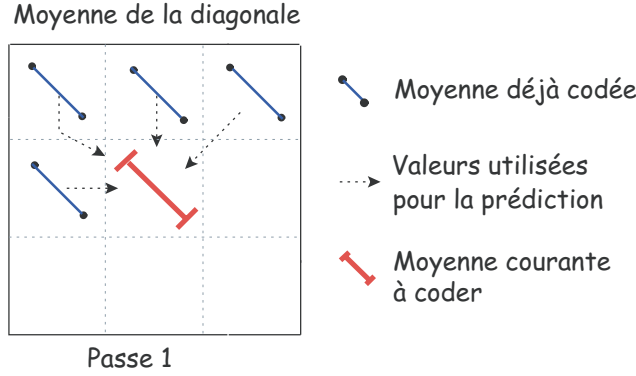


FIG. 5.2: Première passe de l'algorithme de Wu.

### 2.3 Deuxième et troisième passes : exploitation du contexte enrichi.

Les deuxième et troisième passes visent à tirer profit des informations contextuelles présentes aussi bien dans le domaine causal (**prédiction intra-niveau**) que dans l'image  $\mu$  sous-échantillonnée définie au paragraphe 2.2 (**prédiction inter-niveau**). La figure 5.3 illustre le contexte employé pour prédire les pixels issus de l'une de ces deux passes.

La deuxième passe a pour but d'encoder la moitié des pixels de l'image : il s'agit des pixels diagonalement adjacents décrits au paragraphe 2.2. Notons  $\check{I}(2x, 2y)$  la valeur estimée du pixel en haut à gauche d'un bloc  $2 \times 2$ , précédemment utilisé pour le calcul de la moyenne  $\mu(x, y)$  de la diagonale de ce bloc.

$$\begin{aligned} \check{I}(2x, 2y) = & 0.9\mu(x, y) + \frac{I(2x+1, 2y-1) + I(2x-1, 2y-1) + I(2x-1, 2y+1)}{6} \\ & - 0.05(I(2x, 2y-2) + I(2x-2, 2y)) - 0.15(\mu(x, y+1) + \mu(x+1, y)). \end{aligned} \quad (5.3)$$

La reconstruction du second pixel de la diagonale  $I(2x+1, 2y+1)$  est alors très simple : connaissant la moyenne de cette diagonale ( $\mu(x, y)$ ), ainsi que la valeur de l'autre pixel, il suffit d'un bit supplémentaire  $\delta_{x,y}$  pour reconstruire  $I(2x+1, 2y+1)$  suivant l'expression

$$I(2x+1, 2y+1) = 2\mu(x, y) - I(2x, 2y) + \delta_{x,y}, \quad (5.4)$$

où  $\delta_{x,y} = (I(2x, 2y) + I(2x+1, 2y+1)) \text{ modulo } 2$ .

Il est important de noter que  $\delta_{x,y}$  est systématiquement égal à zéro lorsque la compression désirée est une compression avec pertes. En effet, un coefficient de quantification minimale égal à 2 suffit pour supprimer ce bit. Par ailleurs, il est évident que l'estimation de  $I(2x, 2y)$  est réalisée

de façon plus précise qu'une méthode à contexte restreint ne pourrait le faire : il en résulte une diminution notable de l'entropie globale des erreurs de prédiction encodées.

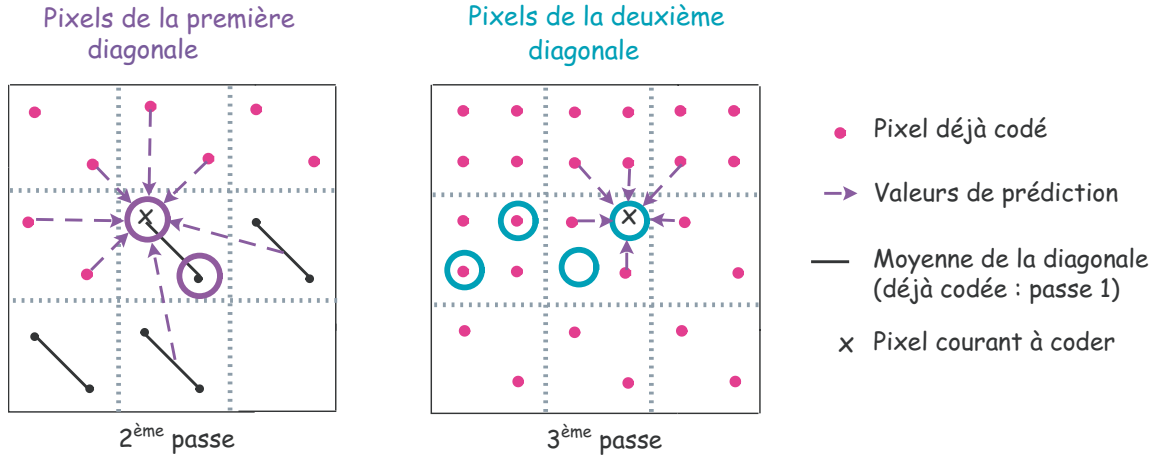


FIG. 5.3: Deuxième et troisième passes de l'algorithme de Wu.

La troisième passe permet l'encodage du reste des pixels (moitié de l'ensemble des pixels de l'image originale), composé des deux autres sites (en haut à droite et en bas à gauche) d'un bloc  $2 \times 2$  donné. Grâce aux étapes précédentes, le contexte adjacent disponible entoure spatialement complètement le pixel à traiter (voir fig.5.3) : les quatre voisins 4-connexes et deux des voisins 8-connexes sont alors connus. La prédiction est alors obtenue par

$$\check{I}(x, y) = \frac{3}{8} (I(x, y-1) + I(x-1, y) + I(x, y+1) + I(x+1, y)) - \frac{I(x-1, y-1) + I(x+1, y-1)}{4}, \quad (5.5)$$

avec  $(x, y) \in \{(2i+1, 2j)\} \cup \{(2i, 2j+1)\}$ ,  $0 \leq i \leq N_x/2$ ,  $0 \leq j \leq N_y/2$ .

### 3 L'approche pyramidale prédictive : principes généraux

Si on considère l'image, ainsi que le propose TMW [MT97], comme étant la superposition d'une information globale (supportant valeur moyenne du signal et contours) et de la texture, il est alors possible de décider, à partir d'un même schéma d'encodage, de fournir une image compressée avec pertes (première partie du message, ou encore information globale), ou sans perte (deuxième partie, correspondant à l'image d'erreurs, autrement dit la texture). Le principe de l'encodage en deux couches de la méthode LAR originelle est donc conservé. La première couche (codeur spatial) se définit par une Approche Pyramidale Prédictive (APP) afin de produire l'image LAR basse résolution (image *LR*). La seconde couche permet de coder la texture avec ou sans perte. A ce niveau, l'utilisateur peut choisir le type de codage appliqué à l'information de texture : encodage classique par méthode spectrale ou par extension de la méthode APP précédemment définie. Le schéma-bloc 5.4 reprend les étapes de la compression.

Ce schéma de compression nécessite tout d'abord d'avoir à disposition une méthode de décomposition pyramidale (construction *Bottom-Up*, §4). Le cheminement à travers la pyramide

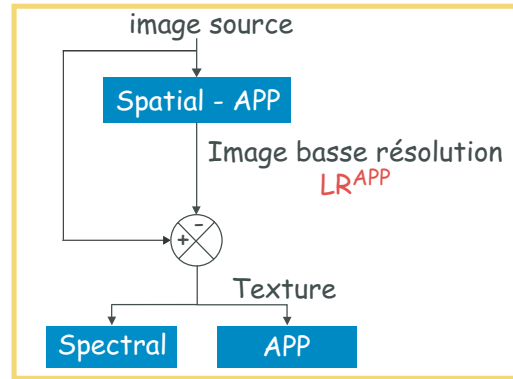


FIG. 5.4: Schéma général du codeur LAR : deux couches de codage, avec deux types de codeur pour la texture.

(descente), utilisé à des fins de compression, s'effectue alors en deux temps : obtention de l'image des blocs puis de la texture superposable à cette image des blocs (paragraphe 5).

La figure 5.5 présente les différentes imagettes successives obtenues par notre approche pyramidale prédictive utilisée pour le codage des deux couches. On peut y voir les étapes d'obtention de l'image LAR dite basse résolution, notée  $LR^{APP}$ , ainsi que le raffinement progressif de l'image. La pyramide est ici complète afin de retrouver après l'opération de décodage l'image originale (compression sans perte).

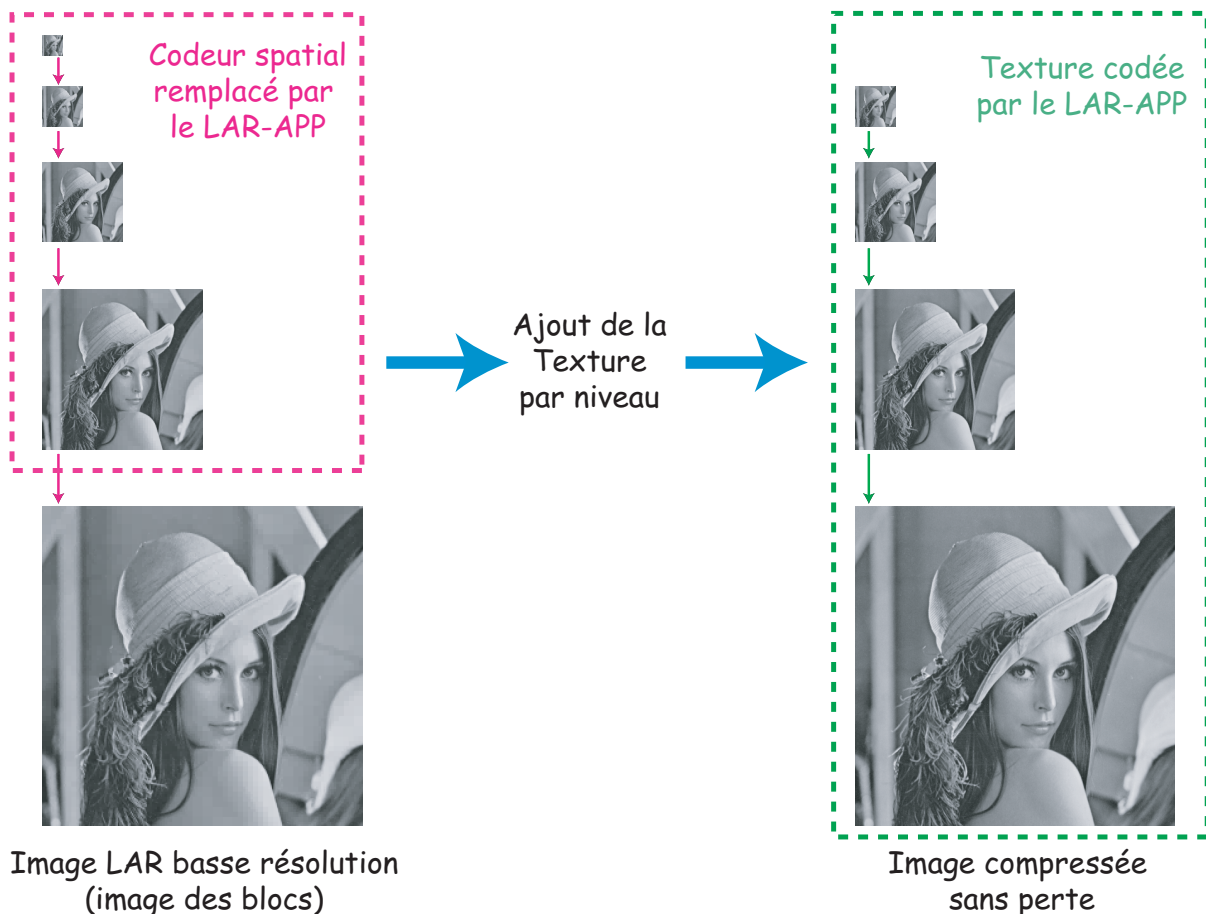


FIG. 5.5: Décomposition pyramidale et compression sans perte : images successives obtenues.



## 4 Construction de la pyramide *Bottom-Up*

Afin d'obtenir une représentation pyramidale de l'image, nous nous plaçons dans le cadre d'une construction d'une pyramide d'images résiduelles (voir chapitre 3), intercalant des phases de prédiction/calcul d'erreurs entre les phases de sous-échantillonnages.

Après quelques remarques préliminaires, permettant de resituer le contexte de notre étude, la seconde partie de cette section établit les équations de construction de la pyramide complète. La troisième partie décrit l'adaptation du principe général de la méthode à notre schéma de compression LAR.

### 4.1 Remarques préliminaires

La représentation multirésolution de notre schéma est donnée par l'ensemble des images  $\{Y_l\}_{l=0}^{l_{max}}$ , avec  $l_{max}$  niveau supérieur de la pyramide, et  $l = 0$  le niveau correspondant à la pleine résolution de l'image. En pratique, on considère le plus souvent que le niveau supérieur de la pyramide correspond aux blocs de taille maximale autorisée ( $16 \times 16$ ), et dans ce cas,  $l_{max} = \log_2(N_{max}) = 4$ .

Rappelons à cet effet un point essentiel : le codage LAR est basé sur l'élaboration préalable d'une partition régulière Quadtree de l'image traitée, à partir de laquelle l'image des tailles en tout point notée  $Siz$  est construite. Le codeur spatial simple produit une image des blocs  $LR$ , où chaque pixel  $LR(x, y)$  est représenté par la valeur

$$LR(x, y) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{m=0}^{N-1} I \left( \left\lfloor \frac{x}{N} \right\rfloor \times N + k, \left\lfloor \frac{y}{N} \right\rfloor \times N + m \right)$$

avec  $N = Siz(x, y)$ .

La première couche du codeur LAR-APP (codeur spatial LAR-APP) se doit elle aussi de créer une image des blocs, représentation de l'image originale à un niveau élevé de scalabilité en distorsion. Cette image est dénommée  $LR^{APP}$ .

Enfin, pour l'encodage de la texture, l'utilisateur doit garder le choix entre application du codeur spectral classique, et l'emploi du codeur de texture LAR-APP.

### 4.2 Construction indépendante de la grille : méthode générale

Le prédicteur de Wu recommande l'application de la passe 1 sur l'image originale, sous-échantillonnée par deux sur chacune des deux dimensions. Chaque pixel de l'image sous-échantillonnée correspond à la moyenne de la diagonale du bloc  $2 \times 2$  dont il est issu. La construction de notre pyramide résulte simplement de l'extension du principe édicté par Wu. Ainsi, chaque pixel des imagerie successives est calculé comme étant la moyenne de la première diagonale du bloc  $2 \times 2$  du niveau précédent. On peut donc écrire, d'après 5.1, l'expression récurrente suivante

$$\begin{cases} l = 0, & Y_0(i, j) = I(i, j); \\ l > 0, & Y_l(i, j) = \left\lfloor \frac{Y_{l-1}(2i, 2j) + Y_{l-1}(2i+1, 2j+1)}{2} \right\rfloor, \end{cases} \quad (5.6)$$

avec  $0 \leq i \leq N_x^l$ ,  $0 \leq j \leq N_y^l$ , pour  $N_x^l = N_x/l$  et  $N_y^l = N_y/l$  (notations identiques au chapitre définissant la méthode LAR existante).

Le sens des indices  $i$  et  $j$  est illustré figure 5.6.

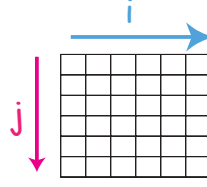


FIG. 5.6: Indices  $i$  et  $j$ .

Cependant, la définition générale de la construction de notre pyramide doit s'affiner, afin de répondre aux contraintes inhérentes à la méthode LAR. Il est en effet indispensable de réaliser une distinction entre compression de la texture par le codeur spectral classique, ou par l'approche pyramidale, objet de ce chapitre.

### 4.3 Prise en compte du pavage LAR - Encodage de la texture

La première étape du codage LAR d'une image consiste en l'obtention d'une image "basse résolution" (en termes de qualité visuelle). Comme indiqué précédemment, cette phase est subordonnée à la création d'un partitionnement Quadtree de l'image originale. Ainsi, pour toute partition Quadtree  $QP^{[N_{max} \dots N_{min}]}$ , on peut associer la valeur d'un bloc de la partition à une valeur issue de la pyramide. Si  $N = Siz(x, y)$ ,  $0 \leq l \leq l_{max}$ , et  $n = 2^l$ , alors on a

$$LR^{APP}(x, y) = Y_l(i, j) \quad | \quad i = \lfloor \frac{x}{n} \rfloor, j = \lfloor \frac{y}{n} \rfloor. \quad (5.7)$$

A ce stade, il existe deux principes de construction de cette image des blocs. La texture pouvant être indifféremment encodée par la couche spectrale ou par le LAR-APP (voir chapitre sur le codeur LAR simple, et schéma 5.4), le codeur doit adapter la construction de la pyramide en fonction des outils choisis par l'utilisateur.

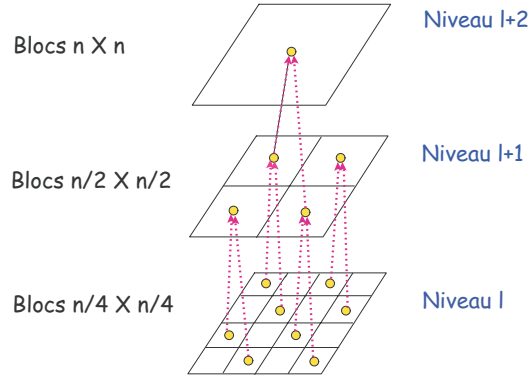
#### 4.3.1 Encodage de la texture par le LAR-APP

**Cette solution implique l'utilisation du même processus de descente sur l'ensemble de la pyramide :** le principe de construction de la pyramide est alors uniforme. A un point du niveau  $l$  de la pyramide précédemment décrite, on fait correspondre un bloc de taille  $2^l \times 2^l$  de l'image originale. Ainsi, la valeur d'un bloc quelconque de la pyramide correspond à la moyenne de la diagonale entière de l'image pleine résolution, aux troncatures successives près (figure 5.7).

On peut donc écrire la relation suivante :

$$LR^{APP}(x, y) = \sum_{k=0}^{\frac{N}{2}-1} \frac{1}{2} \left( I \left( \left\lfloor \frac{x}{N} \right\rfloor \times N + 2k, \left\lfloor \frac{y}{N} \right\rfloor \times N + 2k \right) + I \left( \left\lfloor \frac{x}{N} \right\rfloor \times N + 2k + 1, \left\lfloor \frac{y}{N} \right\rfloor \times N + 2k + 1 \right) \right), \quad (5.8)$$

avec  $N = Siz(x, y)$ .

FIG. 5.7: Construction *bottom-up*.

L'image LAR-APP des blocs engendrée est donc différente de celle produite par le schéma de base (un bloc est initialement reconstruit par la moyenne des pixels le composant). Cependant, la perception visuelle de ces différences se révèle très difficile. Ceci est probablement dû au fait que le pavage LAR s'établisse après observation de l'activité locale de l'image, afin de construire des blocs homogènes. L'image  $LR^{APP}$ , présentée ci-dessus, donne une représentation par la valeur de la diagonale, très proche d'une représentation par la valeur moyenne. En effet, la dispersion des valeurs des pixels est faible pour un bloc situé dans les zones uniformes (typiquement pour les blocs de taille supérieure à  $N_{min} \times N_{min}$ ). Pour les blocs de taille minimale, l'œil est moins sensible aux variations de luminances qu'il ne l'est sur des grands blocs : à nouveau, les différences engendrées par l'approche pyramidale sont peu perceptibles.

#### 4.3.2 Encodage de la texture par le codeur spectral

Cette fois-ci, le codeur spatial LAR-APP se doit de respecter les principes initiaux : la valeur des blocs de l'image "basse résolution" se calcule par la moyenne de chacun de ces blocs. Cette contrainte est nécessaire pour une reconstruction de la texture par l'application de DCT à tailles de blocs variables, la valeur des blocs de l'image "basse résolution" correspondant au coefficient  $DC$ .

De ce fait, l'image  $LR^{APP}$  équivaut exactement à l'image  $LR$  du codeur LAR simple :

$$LR^{APP}(x, y) = LR(x, y), \quad (5.9)$$

$\forall (x, y)$  tels que  $0 \leq x \leq N_x$  et  $0 \leq y \leq N_y$ .

La construction de la pyramide est donc conditionnée suivant les relations :

$$\begin{cases} l = 0, & Y_0(i, j) = LR^{APP}(i, j) = LR(i, j), \\ l > 0, \text{ et } Siz(2^l i, 2^l j) < 2^l & Y_l(i, j) = \left\lfloor \frac{Y_{l-1}(2i, 2j) + Y_{l-1}(2i+1, 2j+1)}{2} \right\rfloor, \\ \text{sinon} & Y_l(i, j) = LR^{APP}(2^l i, 2^l j). \end{cases} \quad (5.10)$$

La figure 5.8 illustre cette construction non uniforme de la pyramide.

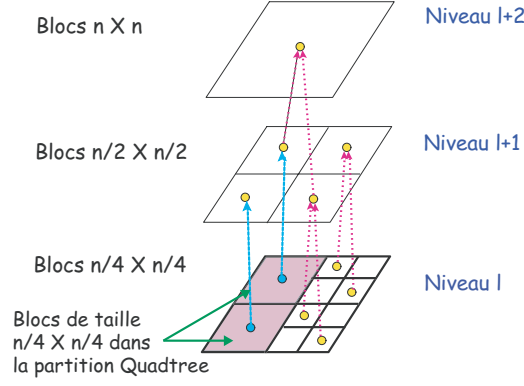


FIG. 5.8: Construction incomplète de la pyramide.

## 5 Descente de la pyramide : exploitation du contexte enrichi

Nous avons donc adopté les principes et les grandes lignes du prédicteur de Wu afin de proposer une descente de la pyramide efficace et à redondance réduite. Afin de simplifier les notations, nous nous plaçons dans le cadre d'une compression sans perte, ce qui implique que les valeurs reconstruites soient exactement égales aux valeurs initiales. De plus, à chaque niveau de la pyramide est associée une taille de bloc. Ainsi, le niveau  $l$  correspond à la taille de bloc  $2^l \times 2^l$ .

### 5.1 Niveau $l_{max}$ de la pyramide

Le niveau supérieur de la pyramide est encodé par un schéma MICD simple, comme le fait la première passe de l'algorithme. Le prédicteur utilisé ici ne correspond pas à celui décrit par Wu : il s'agit plutôt du prédicteur médian MED, implanté dans le standard de compression LOCO-I [WSS96], qui s'est avéré plus efficace pour le dernier niveau de la pyramide. Soit  $\check{Y}_l(i, j)$  la valeur estimée du pixel  $(i, j)$  de l'image  $Y_l$  au niveau  $l$  de la pyramide.

$$\check{Y}_{l_{max}}(i, j) = \begin{cases} \min(Y_{l_{max}}(i-1, j), Y_{l_{max}}(i, j-1)) & \text{si } Y_{l_{max}}(i-1, j-1) \geq \max(Y_{l_{max}}(i-1, j), Y_{l_{max}}(i, j-1)), \\ \max(Y_{l_{max}}(i-1, j), Y_{l_{max}}(i, j-1)) & \text{si } Y_{l_{max}}(i-1, j-1) \leq \min(Y_{l_{max}}(i-1, j), Y_{l_{max}}(i, j-1)), \\ Y_{l_{max}}(i-1, j) + Y_{l_{max}}(i, j-1) - Y_{l_{max}}(i-1, j-1) & \text{sinon.} \end{cases} \quad (5.11)$$

La totalité des pixels contenus dans cette image est encodée par application de ce prédicteur. Les erreurs de prédiction sont donc transmises au décodeur.

### 5.2 Obtention de l'image LAR basse résolution

L'ensemble des opérations suivantes sont conditionnées à la partition réalisée au préalable. Nous avons retenu les tailles de blocs  $N_{min} \times N_{min} = 2 \times 2$  (contours de l'image),  $4 \times 4$ ,  $8 \times 8$  et  $N_{max} \times N_{max} = 16 \times 16$ .

Les niveaux inférieurs sont traités au moyen de deux passes successives, correspondant aux passes 2 et 3 de l'algorithme de WU (paragraphe 2.3). La figure 5.9 permet de visualiser le

comportement de notre algorithme de décomposition *top-down*, en fonction de la taille du bloc indiquée par le partitionnement initial.

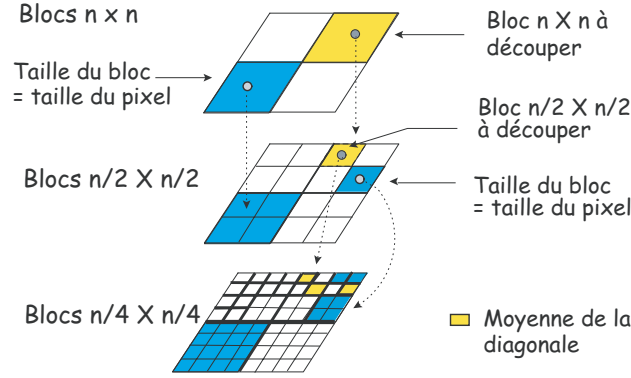


FIG. 5.9: Décomposition Top-Down de la pyramide.

L'idée est la suivante : chaque bloc est divisé jusqu'à ce que sa taille maximale (donnée par la partition Quadtree du codeur spatial) soit atteinte. Ainsi, pour chaque niveau  $l$  donné, et pour chaque pixel de l'imagette  $Y_l$ , on regarde son appartenance à une taille de bloc calculée par le pavage Quadtree. Plusieurs cas de figure sont alors possibles :

1. Si cette taille de pixel est *inférieure ou égale* à la taille du bloc dans la partition Quadtree, i.e.  $Siz(x \times l, y \times l) \leq 2^l$ , alors le pixel en question est traité via la passe 2 ou 3 de l'algorithme (reconstruction des quatre valeurs issues de la décomposition d'un bloc du niveau supérieur).
2. Si la taille des blocs est *supérieure*, à savoir  $Siz(x \times l, y \times l) > 2^l$ , les valeurs sont recopiées afin de raffiner le contexte, ce qui s'écrit (avec les notations introduites au chapitre 1 :

$$Y_l(b^2(i, j)) = Y_{l+1}(i, j). \quad (5.12)$$

Pour la suite de ce paragraphe, nous ne nous intéressons qu'aux seuls blocs nécessitant une décomposition (soit  $Siz(x \times l, y \times l) \leq 2^l$ ). Ainsi, si  $Siz(x \times l, y \times l) \leq 2^l$ , l'estimation des valeurs des pixels  $Y_l(2i, 2j)$  concernés par la passe 2 du prédicteur de Wu (équation 5.3), au niveau  $l$ , peut donc se définir par

$$\begin{aligned} \check{Y}_l(2i, 2j) = & 0.9Y_{l+1}(i, j) + \frac{Y_l(2i+1, 2j-1) + Y_l(2i-1, 2j-1) + Y_l(2i-1, 2j+1)}{6} \\ & - 0.05(Y_l(2i, 2j-2) + Y_l(2i-2, 2j)) - 0.15(Y_{l+1}(i, j+1) + Y_{l+1}(i+1, j)). \end{aligned} \quad (5.13)$$

De même, la reconstruction du second pixel de la diagonale  $Y_l(2i+1, 2j+1)$  du bloc issu de la décomposition d'un bloc au niveau supérieur, peut se réécrire comme étant

$$Y_l(2i+1, 2j+1) = 2Y_{l+1}(i, j) - Y_l(2i, 2j) + \delta_{i,j}^l, \quad (5.14)$$

où  $\delta_{i,j}^l = (Y_l(2i, 2j) + Y_l(2i+1, 2j+1))$  modulo 2 est l'information transmise.

Quant aux pixels traités par la passe 3 (deuxième diagonale d'un bloc  $2 \times 2$ ), l'équation 5.5 est modifiée :

$$\begin{aligned} \check{Y}_l(i, j) = & \frac{3}{8} (Y_l(i, j-1) + Y_l(i-1, j) + Y_l(i, j+1) + Y_l(i+1, j)) \\ & - \frac{Y_l(i-1, j-1) + Y_l(i-1, j+1)}{4}, \end{aligned} \quad (5.15)$$

avec  $(i, j)$  satisfaisant les conditions suivantes :

$$(i, j) \in (2k + \alpha, 2m + \beta),$$

avec  $0 \leq k \leq N_x^l/2, 0 \leq m \leq N_y^l/2$ , et  $(\alpha, \beta) \in (0, 1) \times (1, 0)$ .

Nous pouvons immédiatement vérifier que le niveau 0 de cette décomposition pyramidale partielle permet de retrouver intégralement l'image  $LR^{APP}$  (définie au §4).

## 6 Descente complète de la pyramide - Redondance réduite

L'extension de la méthode à la compression sans perte consiste à établir un codage pyramidal total. Chaque bloc  $N \times N$  de la représentation LAR est ainsi décomposé jusqu'à l'obtention de la pleine résolution, selon notre principe de décomposition pyramidale et de prédiction par contexte enrichi. Le codage sans perte implique simplement l'absence de quantification des erreurs de prédiction à travers la pyramide.

Ainsi, le processus total d'encodage nécessite deux descentes complémentaires successives (voir figure 5.5). Si la première réalise la reconstruction de l'image basse résolution  $LR^{APP}$ , la seconde permet de récupérer la texture. Le niveau 0 de la pyramide est entièrement traité lors de la deuxième descente. Néanmoins, les contours (les blocs de taille  $2 \times 2$ ) sont dissociés de l'information issue des blocs de taille supérieure, de telle sorte que le caractère scalable de l'encodage soit renforcé.

Par ailleurs, dans le cas d'une décomposition pyramidale classique du type laplacien, gaussien ou encore d'images résiduelles, le désavantage majeur est de produire un nombre de symboles codés égal à  $4/3$  du nombre de pixels de l'image pleine résolution. Si les approches du type ondelettes ou pyramides réduites ne présentent pas de redondance apparente, la normalisation introduite sur-dimensionne toutefois les coefficients obtenus : la redondance en termes de bit est diluée sur l'ensemble des valeurs issues de la décomposition (augmentation de la dynamique).

Cette redondance inhérente à la structure de la pyramide à prédiction/résidus est minimisée dans notre schéma. En effet, le surcoût induit par le bit de la première diagonale est limité à 0,25 bit par bloc  $2 \times 2$  et par niveau. De plus, si une quantification est appliquée, ce bit disparaît et le nombre de symboles à transmettre est exactement égal au nombre de blocs du LAR (image basse résolution  $LR$ ) pour la première couche du codeur, ou encore égal au nombre de pixels de l'image originale dans le cadre d'une décomposition complète de la pyramide. Cette propriété rend la méthode efficace en termes d'entropie pour de la compression progressive avec ou sans perte d'une image donnée.

## 7 Réduction de l'entropie - Modélisation implicite du contexte

L'entropie globale est réduite lorsque l'on peut isoler des classes de symboles respectives à différentes lois probabilistes. Ce principe est souvent utilisé et exploité par le biais de la modélisation de contexte [WMS95].

La modélisation de contexte est en effet devenue un facteur clé d'efficacité des algorithmes de compression d'images, en particulier pour les méthodes sans perte : pour preuves les résultats obtenus par les codeurs CALIC [WMS95] et LOCO-I [WSS96]. Ces algorithmes exploitent le fait que l'information de même nature tend à être regroupée spatialement pour chaque sous-bande de l'image. Ainsi, dans un voisinage contenant des coefficients de grande amplitude, rencontrer un autre coefficient de grande amplitude est un événement très probable.

Par ailleurs, le conditionnement réduit l'entropie : celui-ci en devient alors le principe fondamental de la modélisation de contexte. Afin d'éviter d'explorer l'ensemble des contextes possibles (engendrant un espace quasi continu), et donc le phénomène de dilution de contextes, ceux-ci sont généralement réunis en classes.

Une estimation de l'activité locale d'un pixel conduit alors à sa classification dans une loi *a priori*. Dans notre schéma, ces classes vont se définir simplement.

D'une part, l'analyse de l'image basse résolution produite par la première couche du codeur LAR permet de séparer l'ensemble des blocs en deux groupes :

- les zones uniformes (homogènes du point de vue de la texture) caractérisées par une entropie faible, pour tous les blocs  $N \times N$  où  $N \in \{4, 8, 16\}$ ,
- les contours (blocs  $2 \times 2$ ), caractérisés par une forte entropie.

Par ailleurs, la méthode de prédiction utilisée induit naturellement une classification des valeurs d'erreurs en fonction de la passe qui les a produites. En effet, les valeurs issues de la passe 2 diffèrent de façon significative de celles de la passe 3. De plus, le bit supplémentaire provenant de la moyenne de la diagonale est transmis séparément, réduisant ainsi considérablement l'entropie globale des symboles transmis.

En résumé, après observation du comportement de notre méthode vis-à-vis de l'entropie créée, la séparation des lois de probabilité des symboles encodés peut s'effectuer à deux niveaux :

- **par la taille des blocs,**
- **par les différentes passes pour chaque niveau de décomposition.**

Niveau	Taille des blocs	Entropie (bpb)	
4	Tous blocs	Passe 1 : 6.06	
		Passe 2	Passe 3
3	Texture : $16 \times 16$	3.48	3.16
	LAR : $8 \times 8, 4 \times 4, 2 \times 2$	5.54	5.57
2	Texture : $16 \times 16, 8 \times 8$	3.29	3.25
	LAR : $4 \times 4, 2 \times 2$	5.27	5.47
1	Texture : $16 \times 16, 8 \times 8, 4 \times 4$	3.23	3.58
	LAR : $2 \times 2$	5.17	5.43
0	Texture : $16 \times 16, 8 \times 8, 4 \times 4$	3.18	3.62
	Contours : $2 \times 2$	4.59	4.83

TAB. 5.1: Codage sans perte de Lena (4.44 bpb -  $T_h = 20$ ) : entropie (en bpb) par niveau de pyramide, par passe et par taille des blocs (bit de la moyenne exclu).

Le tableau 5.1 illustre la différenciation exercée pour la transmission en sous-flux des erreurs de prédiction et le calcul de l'entropie. La dénomination "LAR" signifie : symboles codant pour l'image basse résolution (image des blocs) LAR lors de la première descente. La texture est quant à elle issue de la seconde descente. Enfin, comme nous l'avons indiqué au paragraphe 6, au niveau 0 de la pyramide, nous différencions les contours (blocs de taille minimale) de la texture (blocs de taille supérieure).

## 8 Robustesse du schéma - Influence du paramètre de seuil

Dans le cadre de la compression sans perte, nous nous sommes intéressés au comportement du système vis-à-vis de la variation du paramètre  $Th$ , seuil appliqué au gradient morphologique pour le calcul de l'image des tailles, défini au chapitre 1. La figure 5.10 illustre l'influence de ce paramètre sur le taux de compression sans perte d'une image donnée (en l'occurrence "Lena").

Nous constatons immédiatement que pour des valeurs raisonnables de ce seuil, le débit binaire reste quasiment constant. Cette observation se reproduit sur un grand ensemble d'images test. Il est ainsi facile d'en déduire la propriété suivante : **le schéma de codage est robuste aux variations de  $Th$** . Notons toutefois que les valeurs possibles de ce paramètre doivent être bornées, afin de conserver une décomposition basée contenu, et donc une modélisation de contexte cohérente - pour un seuil trop faible, la grille n'est composée que de petits blocs, et à l'inverse, une valeur de seuil trop élevée induit un pavage régulier à taille de bloc unique.

De ce fait, afin de simplifier l'utilisation de ce codeur, dans le cas d'une compression sans perte,  $Th$  **peut être fixé a priori**<sup>3</sup>. La valeur optimale du seuil implique une partition de l'image telle que l'équilibre visuel s'établisse entre nombre de grands blocs et nombre de petits blocs. En effet, lors d'une génération excessive de petits blocs, le schéma aura tendance à se comporter comme une décomposition en ondelettes entières (de type S). A contrario, le codage s'apparentera à JPEG dès lors qu'un trop grand nombre de grands blocs sera produit. Le test précédent, effectué sur un ensemble d'images test classique, nous permet de situer ce seuil optimal entre les valeurs 15 et 35. Visuellement, les pavages résultants restent cohérents du point de vue de leur contenu.

## 9 Compression sans perte : Résultats et comparaisons

Il s'avère très délicat d'apprécier l'efficacité d'une décomposition multirésolution dans le cadre d'une compression sans perte. En effet, le critère de comparaison devient dans ce cas la taille du flux après encodage de l'image par un procédé réversible. L'état de l'art des méthodes de compression d'images sans perte désigne le codeur CALIC comme étant l'un des plus efficaces en termes d'entropie du premier ordre (taux de compression) ainsi que de complexité [Bab02]. L'algorithme S+P [SP93] constitue quant à lui une référence en matière de schéma de compression progressif. Nous avons donc procédé à des tests comparatifs avec cet algorithme. Dans cette étude, l'encodage entropique (par codage arithmétique adaptatif) n'est pas implanté. La version des codeurs disponibles intégrant par défaut un étage dédié au codage entropique, le premier travail a

<sup>3</sup> $Th$  est un paramètre ajustable global du codeur



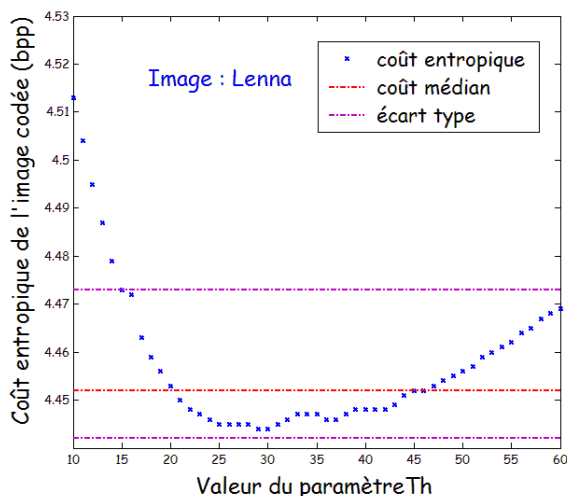


FIG. 5.10: Visualisation de l'influence de la valeur seuil  $Th$  sur la compression sans perte d'une image.

été de réaliser ces encodeurs S+P et CALIC, afin de retrouver les valeurs entropiques du premier ordre.

Le tableau 5.2 affiche les résultats obtenus par notre approche en comparaison avec ceux de CALIC et de S+P, ainsi que le gain entropique conséquent obtenu par le regroupement des symboles encodés en différentes lois (voir §7). Actuellement, les performances sont légèrement en dessous de l'état de l'art : par exemple, CALIC obtient une entropie s'élevant à 4.33 bpp pour l'image correspondant à la composante de luminance de Lena, d'entropie brute 7.45 bpp.

Notons que si le LAR-APP fournit en moyenne un taux de compression comparable, de grandes disparités peuvent être aisément remarquées. Ainsi, on ne peut conclure sur l'efficacité *a priori* d'une méthode par rapport à une autre. Cependant, notre codeur, pouvant fournir une image compressée avec ou sans perte à l'aide du même algorithme, offre la possibilité supplémentaire d'obtenir une **image intermédiaire bas débit d'excellente qualité**. Ces avantages non négligeables en font un candidat tout indiqué aux applications multimedia du type télémédecine, supervision...

## 10 Compression avec pertes : Post-traitement et qualité visuelle

Lors d'une compression avec pertes d'informations, l'enjeu est d'ajuster la phase de quantification de telle sorte que les distorsions introduites soient les moins gênantes possibles. Dans le cadre d'une décomposition pyramidale, la quantification des valeurs transmises est appliquée sur l'ensemble des niveaux de la pyramide. Cependant, pour une compression à bas débit, le bruit de quantification étant cumulatif, la distorsion résultante peut devenir importante lorsqu'elle est propagée le long de la structure pyramidale. Néanmoins, nous nous affranchissons de ce problème en considérant la première descente de la pyramide : grâce au découpage des blocs dépendant de la partition réalisée au préalable, la quantification s'adapte selon la relation 1.8. De ce fait, la distorsion observée reste tout à fait acceptable.

Cependant, les images obtenues par compression avec pertes à l'aide de la méthode pyramidale du LAR possèdent des défauts visibles sur l'image pleine résolution. La première étape de post-traitement consiste à réaliser un filtrage sur les zones uniformes de l'image obtenue, afin d'éliminer

Image	Entropie (bpp)			
	LAR-APP sans partition	LAR-APP	CALIC	S+P
Baboon	6.30	6.13	6.14	6.11
Barb	5.32	5.02	4.93	4.86
Barb2	5.35	5.11	4.93	5.01
Bike	5.45	4.73	4.53	4.67
Cafe	6.26	5.59	5.37	5.56
Gold	4.92	4.85	4.65	4.71
Hotel	5.32	4.85	4.57	4.73
Lena	4.87	4.42	4.33	4.33
Peppers	5.09	4.80	4.58	4.64
tools	6.01	5.65	5.53	5.64
us	4.22	3.66	3.60	3.79
zelda	4.80	4.16	3.98	3.96
Moyenne	5.32	4.91	4.76	4.83

TAB. 5.2: Entropie du premier ordre (bit/pixels) de la méthode Interleaved S+P (avec partition Quadtree ou non), CALIC et S+P.

en particulier les "effets de blocs", artefacts visibles aux frontières de ces blocs (chapitre 1, §2.4). L'obtention de cette pleine résolution à partir des informations issues des résolutions supérieures peut se faire par interpolations successives. Pour ce faire, plusieurs techniques classiques (bicubique, splines,...) ou non (post-traitement efficace élaboré par le laboratoire) ont été testées. Nous avons retenu les travaux réalisés par Darian Muresan : les caractéristiques visuelles des images produites sont meilleures que celles des autres méthodes, en particulier sur les contours des objets. Après de nombreux échanges, une collaboration informelle s'est entamée, jusqu'à ce que Darian Muresan dépose un brevet sur son idée, rendant impossible, pour nous, toute exploration ultérieure (en particulier, par l'indisponibilité du code source).

Muresan [MP01] a mis en place un algorithme d'interpolation directionnelle basée sur la théorie de la récupération optimale adaptative (**optimal recovery**) des valeurs manquantes, élaborée par Golomb [GW59] et appliquée une première fois à l'interpolation par Shenoy et Parks [SP92]. Pour une présentation détaillée de cet outil, le lecteur pourra consulter l'article écrit par Michelli et Rivlin (1977) [MR76], ainsi que celui de Arestov (1989) [Arc89].

Les échantillons locaux manquants de la portion d'image  $I_P$  sont estimés à l'aide des pixels déjà connus de  $I_P$ , en supposant que le "patch" donné  $I_P$  (élément de surface paramétrique) appartient à une classe connue  $K$  de signal quadratique.  $K$  est alors définie par

$$K = \{I_P \in \mathbb{R}^n : I_P^T Q I_P \leq \varepsilon\}.$$

L'estimation via *optimal recovery* des valeurs manquantes se fait par minimisation de l'erreur maximum obtenue sur l'ensemble des vecteurs possibles de  $K$  qui possèdent les mêmes échantillons connus que ceux de notre  $I_P$  donné. L'inverse de la matrice  $Q$  est alors la matrice de covariance des patches locaux de l'image (appelés aussi vecteurs d'apprentissage) [MP05]. Ainsi, la construction d'un filtre d'interpolation est équivalente à minimiser l'erreur quadratique dans le domaine fréquentiel.

Cette nouvelle méthode d'interpolation étant essentiellement basée sur l'apprentissage, l'image basse résolution peut s'utiliser afin d'améliorer le rendu de l'image finale. Enfin, cette approche

tend à préserver les contours, par l'application de l'interpolation le long de ces contours [Mur02].

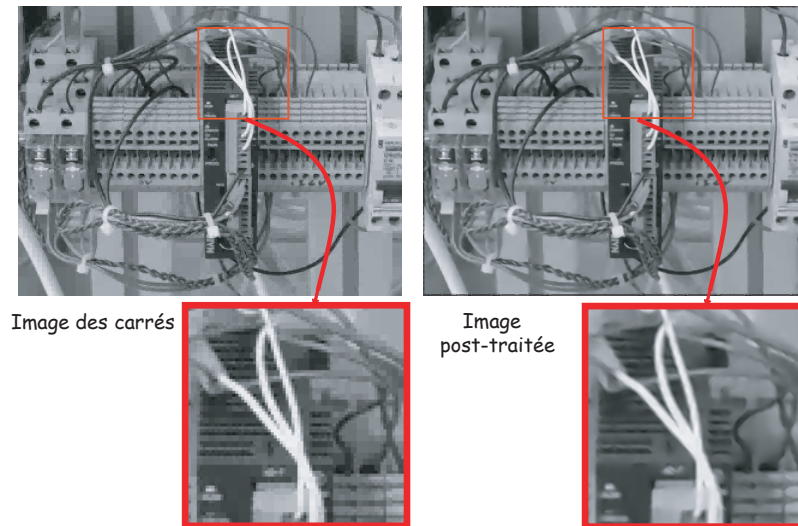


FIG. 5.11: Effet de l'interpolation sur l'image des blocs LAR (image basse résolution - 0.31 bpp) : décomposition selon la première descente jusqu'au niveau 1 pour la luminance, 2 pour les deux composantes de chrominance

Les résultats visuels obtenus par la combinaison de cette interpolation et notre méthode s'affirment réellement très encourageants (voir figure 5.11). Le codage basé contenu du codeur LAR contribue très certainement au bon comportement de l'algorithme détaillé par Muresan, l'information sur les contours étant conservée par le LAR.

Comparée aux standards JPEG (nombreux effets de blocs) et JPEG-2000 (efficace à très bas débit), dans un même contexte et à différents niveaux, la qualité des images produites par le LAR-APP est subjectivement supérieure (fig. 5.12). Les effets dits de rebonds classiques pour JPEG-2000 ne sont pas présents dans notre image.

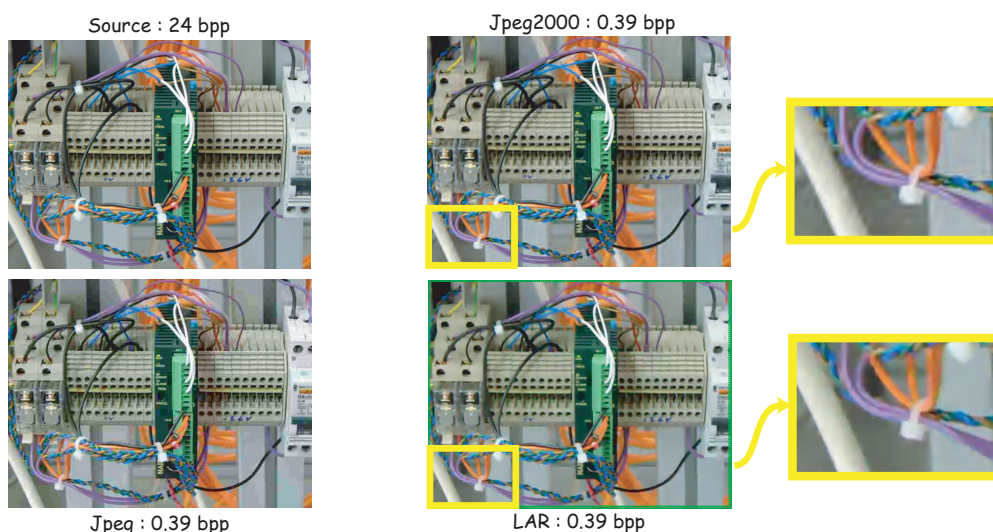


FIG. 5.12: Comparaison des images obtenues après compression à taux égal par les algorithmes JPEG, JPEG-2000 et LAR-APP.

La figure 5.13 propose une autre comparaison visuelle d'une partie de l'image de référence "Akiyo" (première image de la séquence), entre notre méthode, JPEG-2000, le mode "intra" de MPEG4 ainsi que celui de H26L. Si JPEG-2000 est totalement scalable et le LAR-APP scalable par niveaux, H26L et MPEG4 ne le sont pas. Les distorsions introduites par le LAR ne sont pas les mêmes que celles des autres approches : les détails de textures apparaissent atténués, mais d'aspect visuel général très satisfaisant.

La technique de Muresan comporte toutefois un inconvénient : la lourdeur de son temps de calcul. Ainsi, il serait intéressant de ne pouvoir appliquer ce post-traitement uniquement sur les contours de l'image (essentiellement les blocs  $2 \times 2$ ), conjointement à une technique plus économe mais tout aussi efficace sur les zones homogènes.

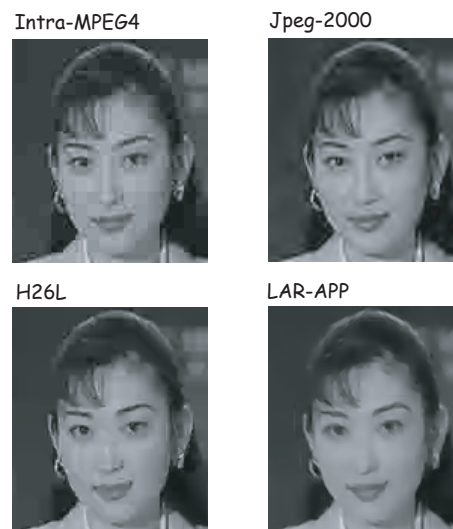


FIG. 5.13: Comparaison des images obtenues après compression à taux égal (0.123 bpp) par les algorithmes Intra-MPEG4, JPEG-2000, H26L (mode d'optimisation maximale) et LAR-APP.

## 11 Conclusion sur l'approche pyramidale prédictive du LAR

Nous avons présenté dans ce chapitre une nouvelle technique originale de compression progressive avec et sans perte, fondée sur la méthode LAR, et que nous avons appelée LAR-APP. Dans cette approche, la seule connaissance de la structure Quadtree est suffisante pour améliorer de façon significative le taux de compression pour le codage sans perte, en regroupant et encodant de façon séparée les informations de même nature.

Par ailleurs, un avantage non négligeable du codeur LAR-APP est de proposer, au moyen d'un unique algorithme, une compression avec ou sans perte. La configuration du codec est de plus simplifiée par le fait que le paramètre utile à la détermination de l'activité locale n'a pas d'impact significatif sur le débit binaire de l'image compressée sans perte. La scalabilité introduite ici élargit le spectre des applications du codec LAR. Les applications de télémédecine, supportant en particulier le codage sans perte de régions d'intérêt à l'intérieur d'une image globalement codée avec pertes, constituent un sujet d'actualité.

Les résultats de compression restent toutefois inférieurs à ceux obtenus par les techniques issus de l'état-de-l'art. Les opérations de prédiction, évoluant dans le domaine spatial uniquement,

conduisent à des performances limitées, en particulier lors de l'encodage de la seconde diagonale (troisième passe). C'est pourquoi nous avons conçu deux schémas alternatifs (chapitres 6 et 7) fondés sur l'association d'une transformée et d'un système de codage prédictif s'exerçant dans le domaine transformé.

## Chapitre 6

# Codage LAR pyramidal prédictif par transformation en S

Le LAR-APP, technique décrite au chapitre précédent (chapitre 5) s'avère être une méthode efficace de codage des images fixes sans perte. Toutefois, même si sa capacité à transmettre un flux scalable par niveau de résolution et par niveau de détails est un atout majeur, les résultats obtenus, en termes d'entropie brute, restent en dessous de l'état de l'art. L'objet de ce chapitre consiste en la présentation d'un schéma nouveau et efficace, directement dérivé du schéma LAR-APP originel.

L'originalité de la méthode mise en œuvre ici tient dans le fait qu'elle allie transformée en S et Approche Pyramidale Prédictive du LAR. Cet **Interleaved S+P**<sup>1</sup> révèle des performances largement supérieures à celles obtenues par le LAR-APP classique, et propose un schéma meilleur, en termes d'entropie globale brute, que CALIC, codeur de l'état de l'art [WMS95]. La structure particulière de l'algorithme proposé ici permet d'envisager la pyramide globale comme étant la résultante de deux pyramides entrelacées. La phase de prédiction avancée fait référence à la méthode classique S+P [SP93]. L'Interleaved S+P fait l'objet d'articles récemment acceptés pour des conférences se tenant en septembre 2005 (ICIP'05 et GRETSI'05) [BDR05b, BDR05a].

La première section présente l'idée générale qui a précédé l'élaboration du Interleaved S+P. La section suivante, quant à elle, décrit le jeu de prédicteurs implantés dans notre schéma. Enfin, l'analyse des résultats découlant de l'application de la méthode sur des images test bien connues permet de conclure de façon objective sur l'efficacité générale du schéma de codage.

## 1 Le LAR-APP : proposition en vue d'améliorer les résultats de compression.

### 1.1 Faiblesses du LAR-APP.

L'Approche Pyramidale a été initialement conçue afin de trouver une façon originale de coder une image par une technique prédictive opérant entièrement dans le domaine spatial. Comme la méthode appartient à la famille des pyramides à prédiction/résidus, une redondance de symboles est nécessairement présente (voir chapitre 5, paragraphe 6).

Par ailleurs, observons les résultats détaillés dans le tableau 5.1. La passe 3 concerne deux fois plus de blocs que la passe 2. Le coût engendré par l'encodage des erreurs de prédiction de

---

<sup>1</sup>Interleaved S+P : S+P entrelacé - Approche Pyramidale Prédictive du LAR associée à la Transformée en S

la passe 3 reste donc important. Il l'est d'autant plus pour les blocs codant l'image LAR basse résolution de détails (image des blocs).

## 1.2 Introduction de la transformée en S dans l'algorithme - Entrelacement

Ces précédentes constatations nous ont conduits à envisager une forme différente de l'Approche Pyramidale Prédictive. Si la structure de l'algorithme est conservée (construction identique de la pyramide, descente en deux passes et différenciation de l'information en fonction de la nature des blocs), nous ne nous restreignons plus désormais à l'exploitation des données uniquement dans le domaine spatial. L'idée est donc la suivante : sur chacune des deux diagonales d'un bloc  $2 \times 2$  donné, issu de la décomposition d'un bloc appartenant à un niveau supérieur de la pyramide, la **transformée en S unidimensionnelle** est appliquée (figure 6.1).

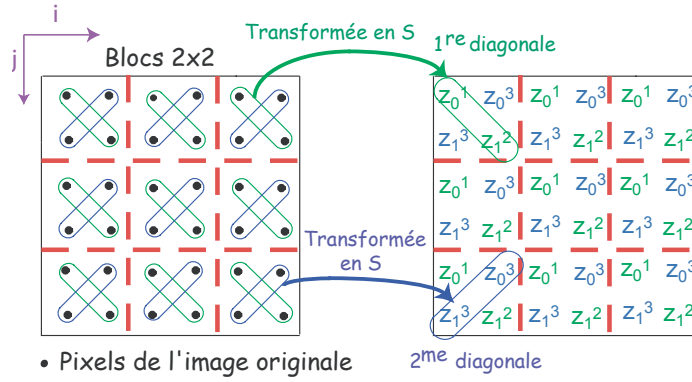


FIG. 6.1: Application particulière de la transformée en S sur une image pleine résolution.

Rappel : la transformée en S directe d'un couple de valeurs  $(u_0, u_1)$  s'écrit :

$$z_0 = \lfloor (u_0 + u_1)/2 \rfloor,$$

$$z_1 = u_1 - u_0.$$

En d'autres termes, les coefficients transformés obtenus correspondent respectivement à la moyenne arrondie à l'entier inférieur des deux valeurs, et au gradient entre ces deux mêmes valeurs. On note  $z_i^k$  ( $i \in \{0, 1\}, k \in \{1, 2, 3\}$ ) les coefficients transformés encodés par la  $k^{\text{ème}}$  passe.

Ainsi, grâce à la construction particulière de notre pyramide, pour la première diagonale d'un bloc  $2 \times 2$  donné, sa valeur moyenne est déjà connue au niveau supérieur. La reconstruction des deux pixels ne nécessite plus que l'obtention, par prédiction, de la valeur du gradient. De ce fait, contrairement au LAR-APP de base, il n'existe pas de redondance de symboles (pas de bit supplémentaire) : l'information redondante inévitable est néanmoins contenue dans le coefficient transformé  $z_1^2$ . En contre-partie, le prédicteur associé doit être suffisamment efficace afin de ne pas augmenter (et si possible diminuer !) la dynamique des erreurs commises. Quant aux pixels concernés par la passe 3 de Wu, la valeur de la moyenne de la deuxième diagonale d'un bloc  $2 \times 2$  ainsi que son gradient subissent à leur tour une phase de prédiction.

L'originalité de l'algorithme Interleaved S+P tient dans le fait qu'il est possible de **décrire la construction de la pyramide entière au moyen de deux pyramides entrelacées** (figure 6.2). La première pyramide en S est construite sur le modèle de celle décrite par le LAR-APP :



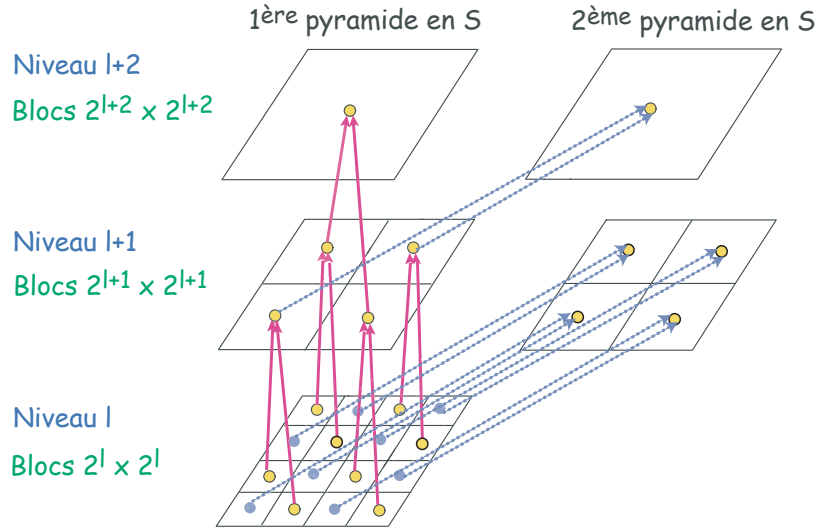


FIG. 6.2: Pyramides entrelacées : principe de construction.

la moyenne de la première diagonale correspond à la valeur du pixel correspondant au niveau immédiatement supérieur. La transformation de la deuxième diagonale d'un bloc  $2 \times 2$  donné peut aussi être vue comme la réalisation d'une seconde pyramide en S : l'estimation de la valeur d'un pixel dépend alors de celles présentes au niveau inférieur de la première pyramide en S. Cette méthode détermine ainsi la notion d'entrelacement de notre algorithme.

Sur le modèle de l'algorithme fondateur S+P [SP93], la méthode Interleaved S+P associe transformée et prédiction du type MICD. Cependant, sa structure particulière, ainsi que la représentation en blocs de taille variable permettent de dépasser très nettement les performances de ce schéma de l'état-de-l'art. Avant de décrire en détail les étapes prédictives, il est nécessaire de définir certains principes généraux régissant l'élaboration de l'ensemble des prédicteurs utilisés.

## 2 Transformée en S et passes de l'algorithme de Wu : Notations et constatations générales

Cette section décrit l'ensemble des notations utiles à la définition des prédicteurs implantés pour l'Interleaved S+P. Afin d'améliorer sensiblement les performances de la méthode, ces prédicteurs se différencient par

- le *coefficient transformé*, correspondant au gradient ou la moyenne,
- la *pyramide en S concernée*, autrement dit la *passé courante*, ou encore l'appartenance d'un pixel donné à la passe 2 ou 3 de l'algorithme de Wu,
- la *nature de l'information* (image LAR "basse résolution" ou texture, voir chapitre 5).

De ce fait, nous obtenons un jeu de prédicteurs localement adaptés au signal traité. De nombreux prédicteurs ont été testés (linéaires, non linéaires, détection de contours...). Ceux répertoriés ici correspondent à l'entropie des erreurs commises la plus faible. Il est certainement possible de trouver encore mieux. Notons cependant le fait que l'expression de ces prédicteurs (§3, §4) a l'avantage d'être relativement simple.



## 2.1 Notations : les coefficients transformés

Nous précisons dans ce paragraphe les notations utilisées dans la suite de ce chapitre. La reconstruction d'un bloc  $2 \times 2$   $Y_l(b^2(i, j))$  du niveau  $l$  de la pyramide s'effectue en deux passes, au travers des deux pyramides entrelacées.

La première phase (première pyramide en S) traite des pixels  $Y_l(2i, 2j)$  et  $Y_l(2i + 1, 2j + 1)$ . La transformée en S appliquée à ce couple de points s'écrit

$$\begin{aligned} z_0^{l,1}(2i, 2j) &= \left\lfloor \frac{Y_l(2i, 2j) + Y_l(2i + 1, 2j + 1)}{2} \right\rfloor, \\ z_1^{l,2}(2i + 1, 2j + 1) &= Y_l(2i, 2j) - Y_l(2i + 1, 2j + 1). \end{aligned} \quad (6.1)$$

De façon similaire, les coefficients transformés de la seconde pyramide en S sont notés

$$\begin{aligned} z_0^{l,3}(2i + 1, 2j) &= \left\lfloor \frac{Y_l(2i, 2j + 1) + Y_l(2i + 1, 2j)}{2} \right\rfloor, \\ z_1^{l,3}(2i, 2j + 1) &= Y_l(2i + 1, 2j) - Y_l(2i, 2j + 1), \end{aligned} \quad (6.2)$$

où les indices  $i$  et  $j$  respectent le sens indiqué sur la figure 5.6.

## 2.2 Cas du coefficient de moyenne - Première pyramide en S

Dans le cas de la passe 2 de Wu, comme il en a été mentionné dans le paragraphe 1.2, le coefficient transformé  $z_0^{l,1}(2i, 2j)$  est donné par la valeur du bloc au niveau supérieur dont il dépend directement. Ainsi, peut-on écrire

$$z_0^{l,1}(2i, 2j) = Y_{l+1}(i, j). \quad (6.3)$$

A un pixel situé au sommet de la pyramide, soit lorsque  $l = l_{max}$ , est associée, de même que pour le LAR-APP, la valeur

$$Y_{l_{max}}(i, j) = z_0^{l_{max}-1,1}(2i, 2j)$$

L'estimation des valeurs de ces pixels est alors réalisée par le prédicteur du LAR-APP (relation 5.11).

Cette section a dégagé l'ensemble des grands principes nécessaires à la mise en place des notations et à la compréhension du lecteur. La section suivante s'attache à définir un jeu de prédicteurs efficaces à l'encodage de l'information issue de l'image des blocs.

## 3 Définition des prédicteurs - Image des blocs

Sur le modèle du LAR-APP, la reconstruction de l'image LAR basse résolution (image des blocs) induit, pour un niveau de la pyramide donné, le traitement des seuls pixels de taille  $Siz(x \times l, y \times l) \leq 2^l$ . Si nous distinguons naturellement la méthode d'estimation des coefficients de moyenne de celle de gradient, nous différencions aussi les techniques de prédiction appliquées sur la première pyramide en S, de celles utilisées pour la deuxième pyramide en S. L'ensemble des prédicteurs décrits ci-après ont été empiriquement déterminés.

### 3.1 Première pyramide en S

La valeur du coefficient  $z_0^{l,1}$  étant déjà connue (niveau supérieur de la première pyramide en S), il nous suffit de procéder à l'estimation du coefficient transformé  $z_1^{l,2}$ , valeur gradient de la première diagonale d'un bloc  $2 \times 2$  donné.

Dans l'objectif d'obtenir l'image des blocs (zones de forte activité), nous avons remarqué qu'une estimation linéaire des coefficients demeure la plus efficace. Pour ce faire, nous nous sommes donc appuyés sur le prédicteur défini par Wu (voir chapitre 5, §2.3). Ainsi, l'estimée des coefficients transformés se calcule par l'équation suivante :

$$\begin{aligned} \hat{z}_1^{l,2}(2i+1, 2j+1) = 2.1 \left[ 0.9Y_{l+1}(i, j) + \frac{Y_l(2i+1, 2j-1) + Y_l(2i-1, 2j-1) + Y_l(2i-1, 2j+1)}{6} \right. \\ \left. - 0.05(Y_l(2i, 2j-2) + Y_l(2i-2, 2j)) \right. \\ \left. - 0.15(Y_{l+1}(i, j+1) + Y_{l+1}(i+1, j)) \right. \\ \left. - Y_{l+1}(i, j) \right]. \end{aligned} \quad (6.4)$$

### 3.2 Deuxième pyramide en S - Image des blocs

La troisième passe de l'algorithme de Wu adapté à notre problématique encode les coefficients transformés issus de la deuxième diagonale d'un bloc  $2 \times 2$ , appartenant de fait à la seconde pyramide en S.

#### 3.2.1 Moyenne de la deuxième diagonale - Premier coefficient

Les prédicteurs appliqués aux valeurs  $z_0^{l,3}$  codant pour l'image LAR basse résolution ou l'image de texture ont la même forme ; seuls des coefficients multiplicateurs diffèrent. L'estimation des coefficients  $z_0^{l,3}$  tire parti des information inter- et intra-niveaux, au moyen de la relation

$$\begin{aligned} \hat{z}_0^{l,3}(2i+1, 2j) = \alpha_0 \frac{Y_l(2i-1, 2j+1) + Y_l(2i, 2j+2) + Y_l(2i+2, 2j) + Y_l(2i+1, 2j-1)}{4} \\ + \beta_0 \hat{z}_0^{l,1}(2i, 2j), \end{aligned} \quad (6.5)$$

avec  $\hat{z}_0^{l,1}(2i, 2j)$  valeur reconstruite, après prédiction, du coefficient  $z_0^{l,1}(2i, 2j)^2$ . L'obtention de l'image des blocs (basse résolution du LAR) s'effectue grâce au couple de valeurs  $(\alpha_0, \beta_0) = \left(\frac{1}{4}, \frac{3}{4}\right)$ .

#### 3.2.2 Gradient de la deuxième pyramide en S - Image des blocs

Après de nombreux essais, nous avons constaté que la meilleure estimation des coefficients  $z_1^{l,3}$  codant l'image basse résolution est obtenue par

$$\begin{aligned} \hat{z}_1^{l,3}(2i, 2j+1) = \alpha_1 (Y_l(2i-1, 2j+1) + Y_l(2i, 2j+2) - Y_l(2i+1, 2j-1) - Y_l(2i+2, 2j)) \\ - \beta_1 (Y_l(2i-1, 2j) + Y_l(2i-1, 2j+2) - Y_l(2i, 2j-1) - \check{Y}_l(2i, 2j+1)), \end{aligned} \quad (6.6)$$

où  $(\alpha_1, \beta_1) = \left(\frac{3}{8}, \frac{1}{8}\right)$ .  $\check{Y}_l(2i, 2j+1)$  correspond à la valeur du pixel estimée  $Y_l(2i, 2j+1)$ , par application du prédicteur de Wu (troisième passe), décrite au chapitre 5, équation 5.15.

---

<sup>2</sup>Dans le cadre d'une compression sans perte,  $\hat{z}_0^{l,1}(2i, 2j) = z_0^{l,1}(2i, 2j)$

## 4 Informations de texture et prédicteurs associés

La réalisation de la deuxième phase de décomposition de la pyramide permet la reconstruction de la texture. Rappelons qu'à chaque niveau de la pyramide, seuls les pixels de taille  $Siz(x \times l, y \times l) > 2^l$  subissent une étape de prédiction, les autres ayant été traités lors du premier processus de descente. Il est de nouveau possible de distinguer le cas des valeurs issues de la première pyramide en S (première diagonale), de celles de la deuxième pyramide en S (seconde diagonale).

### 4.1 Première pyramide en S - Texture

L'information de texture se caractérise par une faible activité locale. Les valeurs de gradient  $z_1^{l,2}$  sont difficilement appréciables à travers un processus linéaire. C'est pourquoi nous nous appuyons sur des opérateurs non linéaires de type médian.

#### 4.1.1 Gradient de la première pyramide en S

Soit  $m_e(u_1, u_2, \dots, u_n)$  la valeur médiane d'un ensemble  $(u_1, u_2, \dots, u_n)$  de  $n$  valeurs. La valeur estimée du coefficient  $z_1^{l,2}(2i, 2j)$  situé dans une zone de texture est donnée par

$$\begin{aligned} \hat{z}_1^{l,2}(2i+1, 2j+1) = \frac{1}{4} & \left( m_e(Y_l(2i-2, 2j), Y_l(2i, 2j-2), Y_l(2i-1, 2j-1)) \right. \\ & \left. + m_e(Y_{l+1}(i+1, j), Y_{l+1}(i, j+1), Y_{l+1}(i+1, j+1)) \right). \end{aligned} \quad (6.7)$$

#### 4.1.2 Remarque : erreur de prédiction sur un coefficient $z_1^l$ d'un bloc de texture

A ce niveau, il est intéressant de rappeler le fait suivant : d'après la construction particulière de la grille de la méthode LAR, les blocs de taille  $4 \times 4$  à  $N_{max} \times N_{max}$  sont constitués de pixels dont la valeur est dans un intervalle d'amplitude  $T_h$ . Ainsi, la valeur prédite de gradient d'un bloc de zone homogène ne peut dépasser, en valeur absolue,  $T_h$ . Ceci vaut aussi bien pour la passe 2 que pour la passe 3 de notre décomposition. De ce fait, un contrôle est effectué après estimation de  $z_1^l$  afin de borner l'erreur commise.

### 4.2 Deuxième pyramide en S - Texture

Il est probable que le contexte intra-niveau très riche disponible pour l'estimation des coefficients transformés  $z_0^{l,3}$  et  $z_1^{l,3}$  suffise pour une prédiction simple linéaire. Ainsi, les valeurs de moyennes  $z_0^{l,3}$  seront traitées par l'application de la relation 6.5, où  $(\alpha_0, \beta_0) = (0.37, 0.63)$ .

Quant aux coefficients  $z_1^{l,3}$ , leur estimation est obtenue après application de l'équation 6.6, avec le couple de valeurs  $(\alpha_1, \beta_1) = \left(\frac{1}{4}, 0\right)$ .

## 5 Compression sans perte : résultats

Pour rendre compte au mieux des performances de la méthodes, nous séparons le cas des images naturelles, évoquées au paragraphe 5.1, des images médicales (§5.3). En outre, nous évaluerons

l'impact de la définition des prédicteurs de l'Interleaved S+P sur les résultats de compression d'images naturelles (§5.3).

### 5.1 Images naturelles : évaluation globale de la méthode - Comparaison avec l'état de l'art.

Afin d'évaluer les performances de l'Interleaved S+P, nous procédons comme il en a déjà été mentionné au paragraphe 9 du chapitre 5. Ainsi, l'étape d'encodage entropique des symboles à transmettre n'a pas été mise en œuvre.

Le tableau 6.1 propose une comparaison des résultats obtenus par application de l'Interleaved S+P, CALIC et S+P sur un ensemble significatif d'images naturelles. Nous remarquons immédiatement le gain significatif apporté par la combinaison de l'Approche Pyramidale Prédictrice du LAR et de la transformée en S. En outre, notre méthode dépasse largement le simple S+P. Les résultats de l'Interleaved S+P **dépassent en moyenne ceux de CALIC**; pour la majorité des images de cette deuxième série de mesures, notre algorithme surpasse CALIC.

Par ailleurs, nous avons évoqué au chapitre précédent, la propriété de robustesse du LAR-APP vis-à-vis de la valeur du paramètre  $T_h$ , seuil du gradient morphologique nécessaire à la construction du sous-échantillonnage non uniforme propre à la méthode LAR. Nous avons également vérifié pour l'Interleaved S+P cette propriété pour la série des images naturelles testées dans le tableau 6.1. Typiquement, la valeur optimale de ce seuil est comprise entre 20 et 40.

Le lecteur aura certainement remarqué le fait suivant : sans la donnée du partitionnement Quadtree, l'Interleaved S+P affiche de moins bonnes performances que son concurrent S+P. De cette constatation, une question se pose : comment se comporte le classique S+P si nous intégrons en plus la phase de différenciation (propre au LAR) des coefficients en fonction de leur appartenance à l'une ou l'autre des descentes pyramidales ? Les premières expérimentations réalisées en ce sens ont montrés que l'entropie résultante reste sensiblement la même lorsque le S+P est associé à une séparation des flux transmis (gain en moyenne de 0.01 bpp). Ce résultat

Image	Entropie (bpp)			
	Interleaved S+P - Sans partition	Interleaved S+P	CALIC	S+P
Barb2	5.21	4.96	4.93	5.01
Cafe	5.66	5.43	5.37	5.56
Gold	4.82	4.64	4.65	4.71
Hotel	4.91	4.66	4.57	4.73
Lena	4.46	4.30	4.33	4.33
Peppers	4.74	4.57	4.58	4.64
tools	5.78	5.52	5.53	5.64
us	4.01	3.55	3.60	3.78
Moyenne	4.95	4.70	4.70	4.80

TAB. 6.1: Entropie du premier ordre de la méthode Interleaved S+P (avec partition ou non), CALIC et S+P.

surprenant mérite une étude plus fouillée, que nous n'avons pu effectuer pour le moment.

## 5.2 Analyse de l'entropie associée aux différentes passes des niveaux de la pyramide

Dans ce paragraphe, nous tentons d'analyser de façon plus fine les données entropiques obtenues. Le tableau 6.2 récapitule le coût entropique des différentes composantes codant l'image "Lena". Il est à mettre en perspective avec les résultats présentés dans le tableau 5.2, au chapitre 5. Enfin, le tableau 6.3 indique le nombre de blocs concernés pour chaque niveau de la décomposition, selon les passes et le coefficient transformé à coder.

Niveau	Taille des blocs	Entropie (bpb)		
4	Tous blocs	Passe 1 : 6.06		
		$z_1^{l,2}$	$z_0^{l,3}$	$z_1^{l,3}$
3	Texture : $16 \times 16$	3.46	2.97	3.13
	LAR : $8 \times 8,4 \times 4,2 \times 2$	6.37	5.10	5.97
2	Texture : $16 \times 16,8 \times 8$	3.83	2.80	3.43
	LAR : $4 \times 4,2 \times 2$	6.17	4.98	5.84
1	Texture : $16 \times 16,8 \times 8,4 \times 4$	3.97	3.08	3.79
	LAR : $2 \times 2$	6.14	4.89	5.78
0	Texture : $16 \times 16,8 \times 8,4 \times 4$	4.02	3.10	3.92
	Contours : $2 \times 2$	5.56	4.23	5.13

TAB. 6.2: Codage sans perte de Lena (4.31 bpb -  $T_h = 20$ ) : entropie (en bpb) par niveau de pyramide, par passe, par coefficient et par taille des blocs.

L'entropie lue pour le niveau 4 de la pyramide (niveau supérieur) est évidemment la même que celle annoncée pour le LAR-APP classique : le traitement des blocs est en effet strictement identique.

Niveau	Taille des blocs	Nb de blocs <sup>a</sup>
4	Tous blocs	Passe 1 : 1024
3	Texture : $16 \times 16$	148
	LAR : $8 \times 8,4 \times 4,2 \times 2$	876
2	Texture : $16 \times 16,8 \times 8$	1464
	LAR : $4 \times 4,2 \times 2$	2632
1	Texture : $16 \times 16,8 \times 8,4 \times 4$	10022
	LAR : $2 \times 2$	6362
0	Texture : $16 \times 16,8 \times 8,4 \times 4$	40088
	Contours : $2 \times 2$	25448

<sup>a</sup>Le nombre de blocs indiqué correspond aux nombres de coefficients  $z_1^{l,2}$  ou  $z_0^{l,3}$  ou  $z_1^{l,3}$ , la passe 1 étant un cas à part.

TAB. 6.3: Nombre de blocs concernés pour chaque niveau, selon la passe et la nature du coefficient (image Lena)

Le gain constaté est dû à une meilleure prédiction des coefficients  $z_0^{l,3}$  et  $z_1^{l,3}$  issus de la passe 3 d'une part, et des coefficients  $z_1^{l,2}$  au niveau de la texture d'autre part. De plus, le regroupement

des données compressées suivant leur nature (image des blocs/texture, passe 2/passe 3, **coefficients**  $z_0^l/z_1^l$ ) et par niveau de décomposition, apporte une fois encore une diminution notable de l'entropie globale brute. Toutefois, les résultats obtenus sur la prédiction des valeurs de gradient de la première diagonale ne sont pas totalement satisfaisants pour les blocs codant l'image "basse résolution" du LAR. C'est certainement sur ce point que des améliorations peuvent être envisagées.

### 5.3 Images médicales

L'imagerie médicale étant une problématique à part dans le domaine de la compression d'images, nous faisons apparaître ici les résultats de l'application de notre algorithme Interleaved S+P sur un ensemble conséquent de données cliniques. Une comparaison directe de ces résultats avec ceux établis par CALIC ou S+P est disponible dans les tableaux 6.4, 6.5, 6.6 et 6.7.

A observer de façon globale l'ensemble des résultats, nous nous apercevons de l'excellent comportement de l'Interleaved S+P vis à vis de ces deux algorithmes de l'état-de-l'art que sont S+P et CALIC. Sur l'ensemble des quatre séries de mesures, **l'entropie d'ordre zéro résultant de l'Interleaved S+P est en moyenne 0.28 bits par pixels plus faible que celle obtenue par CALIC** (lui même meilleur que S+P)! L'écart moyen s'affirme donc considérable, en particulier sur les IRM<sup>3</sup> et les mammographies.

Essayons maintenant d'analyser ce surprenant résultat. Pour cela, on pourra prêter plus d'attention au contenu des images testées et présentées en annexe I. Les images du type IRM et mammographies ont ceci de particulier qu'elles contiennent (pour la plupart) de larges zones uniformes. Pour notre méthode fondée sur une représentation à taille de bloc variable, il devient aisé d'encoder efficacement ce "fond".

Par ailleurs, contrairement à ce qui a été écrit pour les images naturelles, nous remarquons que la valeur du seuil n'est cette fois plus indifférente à la performance (en termes de débit) de l'algorithme : typiquement, la valeur du seuil optimale est ici égale à 2. Ainsi, le "tissu" est représenté intégralement par un pavage régulier de blocs de taille  $2 \times 2$ , alors que le fond est rempli par des carrés de taille maximale. Même si la valeur de ce seuil reste très faible, elle est pourtant essentielle à la bonne tenue de notre algorithme : la séparation des lois entropiques constitue toujours un point crucial de la méthode. Notons que la taille maximale de notre partitionnement a été maintenue, comme dans le cas des images naturelles, égale à  $16 \times 16$  : il serait probablement judicieux, pour ces images spécifiques, de pouvoir augmenter cette taille ( $32 \times 32$  ou  $64 \times 64$ ). Faute de temps, ces expérimentations ne pourront être menées à bien pour le moment.

Pour le reste des images, l'Interleaved S+P se comporte de la même manière que pour les images naturelles. La méthode est globalement **meilleure que CALIC** (et bien sûr S+P), de l'ordre de **0.05 bpp** (images "colon", "angiot", "us", "angio", "echo", "cr", "x\_ray", "ct\_abdomen\_fr" et "ct\_abdomen\_tr").

---

<sup>3</sup>IRM : Image par Résonance Magnétique

Image	Taille	Entropie (bpp)		
		S+P	CALIC	Interleaved S+P
colon	$512 \times 512$	3.53	3.33	<b>2.87</b>
angiot	$512 \times 512$	5.20	5.19	<b>4.99</b>
us	$512 \times 448$	3.79	3.60	<b>3.55</b>

<a href="http://eelmpo.cityu.edu.hk/imagedb/">http://eelmpo.cityu.edu.hk/imagedb/</a>				
angio	$512 \times 512$	3.11	<b>3.10</b>	<b>3.10</b>
cr	$1744 \times 2048$	3.30	3.30	<b>3.27</b>
echo	$720 \times 496$	<b>3.28</b>	3.41	3.44
x_ray	$2048 \times 1680$	2.39	<b>2.27</b>	2.36
Résonance magnétique				
mr_head_p	$256 \times 256$	4.27	4.26	<b>4.08</b>
mr_thorax	$256 \times 256$	<b>4.60</b>	4.71	4.61
Coupes tomographiques				
ct_abdomen_fr	$512 \times 512$	2.61	<b>2.54</b>	2.61
ct_abdomen_tr	$256 \times 256$	3.43	<b>3.38</b>	3.43
Mammographies				
mammo	$2048 \times 2048$	1.74	1.70	<b>1.45</b>
mdb001	$1024 \times 1024$	1.64	1.60	<b>1.26</b>
mdb003	$1024 \times 1024$	1.85	1.78	<b>1.58</b>
mdb005	$1024 \times 1024$	2.19	2.13	<b>1.80</b>
mdb007	$1024 \times 1024$	1.91	1.86	<b>1.53</b>
mdb023	$1024 \times 1024$	2.11	2.04	<b>1.83</b>
mdb028	$1024 \times 1024$	2.14	2.06	<b>1.81</b>
mdb058	$1024 \times 1024$	1.93	1.87	<b>1.50</b>
mdb063	$1024 \times 1024$	1.73	1.67	<b>1.29</b>
mdb076	$1024 \times 1024$	1.82	1.75	<b>1.50</b>

<a href="http://www.cis.rit.edu/htbooks/mri/inside.htm">http://www.cis.rit.edu/htbooks/mri/inside.htm</a>				
Images par Résonance Magnétique				
an01	$512 \times 512$	3.22	3.03	<b>2.75</b>
an02	$512 \times 512$	<b>3.13</b>	3.19	3.30
an06	$512 \times 512$	2.73	2.61	<b>2.53</b>
hd01	$512 \times 512$	3.23	3.17	<b>2.61</b>
hd03	$512 \times 512$	2.91	2.93	<b>2.30</b>
hd08	$512 \times 512$	3.57	3.47	<b>3.08</b>
ex03	$512 \times 512$	<b>3.88</b>	<b>3.88</b>	3.97
nk02	$512 \times 512$	2.26	2.09	<b>1.73</b>
sp09	$512 \times 512$	3.15	3.14	<b>2.97</b>

MOYENNE	2.89	2.83	<b>2.63</b>
---------	------	------	-------------

TAB. 6.4: Comparaisons entre S+P, CALIC et Interleaved S+P sur un ensemble d'images médicales de diverses natures.

<a href="http://eelmpo.cityu.edu.hk/imagedb/">http://eelmpo.cityu.edu.hk/imagedb/</a>					
<b>DENSE-GLANDULAR</b>					
Image	Taille	S+P	CALIC	Interleaved S+P	Seuil
<b>sein normal</b>					
mdb003	1024 × 1024	1.85	1.78	<b>1.58</b>	2
mdb004	1024 × 1024	1.93	1.84	<b>1.64</b>	2
mdb033	1024 × 1024	1.43	1.36	<b>1.12</b>	2
mdb034	1024 × 1024	1.46	1.40	<b>1.14</b>	2
mdb035	1024 × 1024	1.43	1.35	<b>1.12</b>	2
mdb036	1024 × 1024	1.44	1.37	<b>1.15</b>	2
mdb037	1024 × 1024	1.41	1.36	<b>1.09</b>	2
mdb038	1024 × 1024	1.33	1.30	<b>0.93</b>	2
mdb039	1024 × 1024	1.24	1.19	<b>1.90</b>	2
mdb040	1024 × 1024	0.97	0.94	<b>0.65</b>	2
	<b>MOYENNE</b>	1.449	1.389	<b>1.232</b>	
<b>lésion bénigne</b>					
mdb063	1024 × 1024	1.73	1.67	<b>1.29</b>	1
mdb099	1024 × 1024	1.72	1.66	<b>1.26</b>	2
mdb104	1024 × 1024	1.63	1.56	<b>1.27</b>	2
mdb107	1024 × 1024	2.04	2.02	<b>1.60</b>	2
mdb126	1024 × 1024	2.03	1.99	<b>1.60</b>	2
mdb145	1024 × 1024	2.48	2.41	<b>2.16</b>	2
mdb163	1024 × 1024	1.98	1.93	<b>1.59</b>	2
mdb165	1024 × 1024	1.35	1.30	<b>0.97</b>	2
mdb193	1024 × 1024	2.11	2.07	<b>1.77</b>	2
mdb198	1024 × 1024	1.82	1.76	<b>1.49</b>	2
	<b>MOYENNE</b>	1.889	1.837	<b>1.500</b>	
<b>tumeur maligne</b>					
mdb058	1024 × 1024	1.93	1.87	<b>1.50</b>	2
mdb102	1024 × 1024	1.63	1.56	<b>1.37</b>	2
mdb105	1024 × 1024	1.91	1.87	<b>1.54</b>	2
mdb111	1024 × 1024	2.07	2.00	<b>1.76</b>	2
mdb125	1024 × 1024	1.93	1.90	<b>1.46</b>	2
mdb130	1024 × 1024	2.06	2.03	<b>1.61</b>	2
mdb170	1024 × 1024	1.60	1.55	<b>1.11</b>	2
mdb171	1024 × 1024	1.77	1.72	<b>1.38</b>	2
mdb179	1024 × 1024	0.99	0.94	<b>0.69</b>	2
mdb202	1024 × 1024	2.20	2.14	<b>1.87</b>	2
	<b>MOYENNE</b>	1.809	1.758	<b>1.429</b>	

TAB. 6.5: Mammographies : tissus glandulaires denses



<a href="http://eelmpo.cityu.edu.hk/imagedb/">http://eelmpo.cityu.edu.hk/imagedb/</a>					
<b>FATTY</b>					
Image	Taille	S+P	CALIC	Interleaved S+P	Seuil
<b>sein normal</b>					
mdb006	1024 × 1024	2.21	2.14	<b>1.79</b>	2
mdb009	1024 × 1024	2.07	1.96	<b>1.85</b>	2
mdb011	1024 × 1024	1.96	1.88	<b>1.69</b>	2
mdb026	1024 × 1024	2.18	2.11	<b>1.76</b>	2
mdb060	1024 × 1024	1.62	1.56	<b>1.21</b>	2
mdb070	1024 × 1024	2.13	2.04	<b>1.83</b>	2
mdb076	1024 × 1024	1.82	1.75	<b>1.50</b>	2
mdb077	1024 × 1024	2.21	2.15	<b>1.79</b>	2
mdb078	1024 × 1024	1.97	1.91	<b>1.53</b>	2
mdb079	1024 × 1024	1.82	1.73	<b>1.47</b>	2
	<b>MOYENNE</b>	<b>1.999</b>	<b>1.923</b>	<b>1.642</b>	
<b>lésion bénigne</b>					
mdb005	1024 × 1024	2.19	2.13	<b>1.80</b>	2
mdb010	1024 × 1024	1.70	1.64	<b>1.29</b>	2
mdb012	1024 × 1024	2.10	1.98	<b>1.86</b>	2
mdb025	1024 × 1024	2.31	2.26	<b>1.90</b>	2
mdb059	1024 × 1024	1.66	1.58	<b>1.23</b>	2
mdb069	1024 × 1024	2.23	2.16	<b>1.90</b>	2
mdb080	1024 × 1024	1.77	1.70	<b>1.43</b>	2
mdb091	1024 × 1024	1.82	1.74	<b>1.48</b>	2
mdb097	1024 × 1024	2.11	2.07	<b>1.61</b>	2
mdb132	1024 × 1024	2.52	2.43	<b>2.33</b>	2
	<b>MOYENNE</b>	<b>2.041</b>	<b>1.969</b>	<b>1.683</b>	
<b>tumeur maligne</b>					
mdb028	1024 × 1024	2.14	2.06	<b>1.80</b>	2
mdb075	1024 × 1024	1.84	1.75	<b>1.50</b>	2
mdb092	1024 × 1024	1.79	1.72	<b>1.46</b>	2
mdb095	1024 × 1024	1.84	1.81	<b>1.37</b>	2
mdb134	1024 × 1024	2.83	<b>2.75</b>	2.83	4
mdb141	1024 × 1024	2.57	2.52	<b>2.21</b>	2
mdb144	1024 × 1024	2.72	<b>2.61</b>	2.70	4
mdb148	1024 × 1024	2.67	<b>2.59</b>	2.62	6
mdb155	1024 × 1024	2.35	2.30	<b>1.99</b>	2
mdb158	1024 × 1024	1.79	1.75	<b>1.35</b>	2
	<b>MOYENNE</b>	<b>2.254</b>	<b>2.186</b>	<b>1.985</b>	

TAB. 6.6: Mammographies : seins graisseux

<a href="http://eelmpo.cityu.edu.hk/imagedb/">http://eelmpo.cityu.edu.hk/imagedb/</a>					
<b>FATTY-GLANDULAR</b>					
Image	Size	S+P	CALIC	Interleaved S+P	Seuil
<b>sein normal</b>					
mdb007	1024 × 1024	1.91	1.86	<b>1.53</b>	2
mdb008	1024 × 1024	2.16	2.10	<b>1.76</b>	2
mdb014	1024 × 1024	1.86	1.79	<b>1.45</b>	2
mdb016	1024 × 1024	1.73	1.70	<b>1.29</b>	2
mdb018	1024 × 1024	1.43	1.38	<b>1.12</b>	2
mdb020	1024 × 1024	2.16	2.10	<b>1.76</b>	2
mdb022	1024 × 1024	1.83	1.77	<b>1.48</b>	2
mdb024	1024 × 1024	2.03	1.93	<b>1.76</b>	2
mdb029	1024 × 1024	2.02	1.93	<b>1.76</b>	2
mdb031	1024 × 1024	2.17	2.12	<b>1.90</b>	2
	MOYENNE	1.930	1.868	<b>1.581</b>	
<b>lésion bénigne</b>					
mdb001	1024 × 1024	1.64	1.60	<b>1.26</b>	2
mdb002	1024 × 1024	2.00	1.93	<b>1.63</b>	2
mdb013	1024 × 1024	1.76	1.71	<b>1.36</b>	2
mdb015	1024 × 1024	1.68	1.63	<b>1.32</b>	2
mdb017	1024 × 1024	1.40	1.34	<b>1.10</b>	2
mdb019	1024 × 1024	2.23	2.19	<b>1.78</b>	2
mdb021	1024 × 1024	2.16	2.09	<b>1.88</b>	2
mdb030	1024 × 1024	1.81	1.75	<b>1.45</b>	2
mdb032	1024 × 1024	2.23	2.17	<b>1.89</b>	2
mdb081	1024 × 1024	2.27	2.21	<b>1.95</b>	2
	MOYENNE	1.918	1.862	<b>1.562</b>	
<b>tumeur maligne</b>					
mdb023	1024 × 1024	2.11	2.04	<b>1.83</b>	2
mdb072	1024 × 1024	1.30	1.26	<b>0.90</b>	2
mdb090	1024 × 1024	1.61	1.56	<b>1.16</b>	2
mdb115	1024 × 1024	2.33	2.24	<b>2.05</b>	2
mdb117	1024 × 1024	1.95	1.88	<b>1.65</b>	2
mdb120	1024 × 1024	2.09	2.00	<b>1.81</b>	2
mdb124	1024 × 1024	1.68	1.61	<b>1.38</b>	2
mdb181	1024 × 1024	1.73	1.68	<b>1.28</b>	2
mdb186	1024 × 1024	1.62	1.57	<b>1.22</b>	2
mdb209	1024 × 1024	2.04	1.97	<b>1.77</b>	2
	MOYENNE	1.846	1.781	<b>1.505</b>	

TAB. 6.7: Mammographies : tissus glandulaires graisseux

## 5.4 Images composites - images aériennes

Afin de compléter les résultats présentés ci-avant, nous avons réalisé des tests sur des images composites d'une part (images comprenant un texte et une photo), et des images aériennes d'autre part. Les entropies observées sont consignées respectivement dans les tableaux 6.8 et 6.9.

Image	Taille de l'image	Entropie (bpp)			
		Interleaved S+P - Sans partition	Interleaved S+P	CALIC	S+P
cmpnd1	512 × 768	2.91	2.15	2.93	2.53
cmpnd2	1024 × 1392	3.18	2.41	3.30	2.87
	MOYENNE	3.04	2.28	3.11	2.70

TAB. 6.8: Entropie du premier ordre (bit/pixels) de la méthode Interleaved S+P (avec partition Quadtree ou non), CALIC et S+P - Images composites

Nous constatons immédiatement le gain conséquent obtenu par notre méthode sur les algorithmes CALIC et S+P (débit moyen à hauteur de 2.28 bpp, soit une entropie 0.42 **bpp plus faible que pour CALIC!**). Dans le cas des images composites, nous retrouvons les caractéristiques évoquées pour les images médicales, à savoir : le fond (blanc), totalement uniforme et donc représenté par des blocs de taille maximale, prend une grande part de l'image. De plus, notre algorithme s'adapte parfaitement à la forme du texte présent dans l'image. De nouveau, le découpage en différents flux permet d'améliorer les performances de l'Interleaved S+P. Remarquons toutefois que, pour cette série, le codeur sans partitionnement préalable fournit également un encodage plus efficace que l'état-de-l'art.

Image	Taille de l'image	Entropie (bpp)			
		Interleaved S+P - Sans partition	Interleaved S+P	CALIC	S+P
JPEG Continuous-tone Test Image Set					
Aerial2	720 × 1024	4.92	4.36	4.63	4.67
Band1	736 × 736	4.24	3.81	3.95	4.00
Band2	736 × 736	4.17	3.67	3.88	3.93
Band3	736 × 736	4.03	3.52	3.73	3.77
Band4	736 × 736	4.52	3.91	4.20	4.23
ftp ://ftp.ipl.rpi.edu/stills/aerial/					
Image1	512 × 512	4.86	4.72	4.74	4.78
Image2	512 × 512	3.50	2.70	3.36	3.37
Image3	512 × 512	4.60	4.57	4.53	4.52
Image4	512 × 512	6.20	6.17	6.17	6.13
Image5	512 × 512	6.24	6.13	6.20	6.15
Image6	512 × 512	6.03	5.84	5.92	5.91
Image7	512 × 512	4.74	5.66	5.66	5.67
	MOYENNE	4.92	4.59	4.74	4.75

TAB. 6.9: Entropie du premier ordre (bit/pixels) de la méthode Interleaved S+P (avec partition Quadtree ou non), CALIC et S+P - Images aériennes

Quant aux images aériennes, nous vérifions de nouveau la supériorité de l'Interleaved S+P face aux méthodes CALIC et S+P. Le gain demeure ici aussi très net : 4.59 bpp pour l'Interleaved contre "seulement" 4.74 bpp pour CALIC en moyenne (soit 0.15 **bpp de mieux**).

## 6 Conclusion - Interleaved S+P

Dans ce chapitre, nous avons présenté une méthode dénommée Interleaved S+P combinant l'approche pyramidale prédictive du LAR et la transformée en S. Cette association originale d'une ondelette fréquemment utilisée et d'une décomposition spatiale particulière permet de construire un schéma de codage en sous-bandes extrêmement efficace tant du point de vue du coût entropique, que de la perception visuelle des images intermédiaires.

En résumé, le Interleaved S+P intègre la famille des décompositions en sous-bandes. De ce fait, aucune redondance de symboles n'est à déplorer, au contraire du schéma LAR-APP classique (redondance réduite à un unique bit). Les images intermédiaires (basse résolution en termes de qualité), exactement égales à celles du LAR-APP, conservent leurs excellentes propriétés psychovisuelles. Par ailleurs, la scalabilité intrinsèque de la méthode proposée constitue un atout essentiel, vis-à-vis des schémas traditionnels du type CALIC.

Sur un ensemble conséquent d'images naturelles, nous avons pu constater les très bonnes performances - en termes d'entropie d'ordre zéro - de l'Interleaved S+P : cette méthode réalise de meilleurs taux de compression que CALIC ou S+P, méthodes établies de l'état-de-l'art.

Les plus spectaculaires avancées ont été évaluées dans le domaine des images médicales. 110 images ont été testées : **l'Interleaved S+P surpasse très largement l'état-de-l'art en compression sans perte !**

Pour le moment, nos résultats sont donnés sous la forme d'entropie brute (entropie d'ordre zéro). Nous avons à disposition un codeur arithmétique adaptatif, et une version du codeur entropique Golomb-Rice [Déf04], implanté en particulier dans nos applications de type "client-serveur" (voir chapitre 9). Des travaux futurs devront porter sur la recherche d'un codeur entropique mieux adapté à notre solution de codage.

Enfin, la robustesse de notre algorithme vis-à-vis d'une mauvaise de récupération des valeurs de grille ou d'erreur de prédiction sera évaluée dans le chapitre 8.



## Chapitre 7

# La transformée RWHT+P

Partant de la structure pyramidale construite pour le LAR-APP (chapitre 5), nous avons envisagé deux solutions de codage. La première a conduit à l'élaboration de l'Interleaved S+P, algorithme présenté au chapitre 6. La seconde méthode, exposée dans ce chapitre, se base sur une exploitation particulière de la transformée Walsh-Hadamard.

La transformée Walsh-Hadamard (WHT<sup>1</sup>) fait partie des techniques qui ont souvent été utilisées à des fins de compression d'images. De nombreuses méthodes multirésolution font appel à cette transformée appliquée typiquement sur des blocs  $2 \times 2$  : des schémas ondelettes réversibles notamment s'appuient sur la WHT. L'encodage sans perte d'une image s'effectue classiquement par l'application d'une version modifiée de la WHT unidimensionnelle, à savoir la transformée en S (voir chapitre 4, §3.1.1). Cette transformée en S s'impose désormais comme une des meilleures bases d'ondelettes entières existantes dans le domaine de la compression sans perte [Abh02].

Il n'existe pas dans la littérature de solution de codage réversible résultant de l'adaptation de la transformée Walsh-Hadamard 2D  $2 \times 2$  traditionnelle. Seule la transformée Walsh-Hadamard réversible de support supérieur à 2 a été étudiée par plusieurs auteurs [PD97, KS01] : les résultats s'avèrent néanmoins moins bons que ceux réalisés par l'application de la transformée en S.

D'autre part, la prédiction des coefficients dans des espaces transformés a suscité également un certain nombre de travaux. En particulier, la méthode RWTP [ICB<sup>+</sup>00] s'appuie sur une transformée en ondelettes sans perte associée à une prédiction sans perte multiéchelle. Les résultats annoncés figurent parmi les tous meilleurs. Cependant, la phase de prédiction est implantée par l'intermédiaire d'un filtre optimisé : les coefficients de ce filtre ont été préalablement obtenus par apprentissage sur l'image selon la méthode du LS-AR (Least Square Auto Regression). La complexité importante de cette technique est un frein majeur à son développement.

Dans ce chapitre, nous nous intéressons donc tout d'abord à la définition de la  $RWHT_{2 \times 2}$  (Reversible  $WHT_{2 \times 2}$ ) 2D, appliquée directement sur des blocs  $2 \times 2$ , et fondée sur l'exploitation d'une fonction de parité (§1). La construction de la pyramide RWHT correspondante est décrite au paragraphe 2.

Par la suite, nous définissons une méthode pyramidale extrêmement efficace de compression sans perte (dénommée "**RWHT+P**<sup>2</sup>"), surpassant largement les méthodes de l'état de l'art et l'Interleaved S+P (chapitre précédent). L'élaboration de l'algorithme se fait par l'association de la RWHT précédemment évoquée et des trois éléments suivants :

---

<sup>1</sup>WHT : Walsh-Hadamard Transform

<sup>2</sup>RWHT+P : Reversible Walsh Hadamard Transform + Prediction

- une **technique de prédiction et d'interpolation**, décrite en section 3,
- un **partitionnement Quadtree**, méthode déjà développée et utilisée aux chapitres 5 et 6,
- une **classification de contexte** réalisée dans l'espace transformé, exposée en section 4.

Enfin, nous vérifions au paragraphe 5 que la combinaison de l'ensemble de ces éléments crée un algorithme extrêmement efficace en termes d'entropie pour de la compression sans perte. Il est à noter que la méthode a été essentiellement développée par Olivier Déforges, et a été une première fois évoquée dans le document [Déf04] : elle fait actuellement l'objet de soumissions en cours d'évaluation.

## 1 Transformée RWHT

La transformée Walsh-Hadamard unidimensionnelle classique s'avère non réversible et, appliquée sur un bloc de taille  $2 \times 2$ , s'exprime par

$$\text{Matrice } WHT_{2 \times 2} := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (7.1)$$

La transformée en S, version réversible de la WHT, diffère de celle-ci par la normalisation opérée selon

$$\text{Matrice } S := \begin{bmatrix} \frac{1}{2} & 1 \\ \frac{1}{2} & -1 \end{bmatrix}. \quad (7.2)$$

Ainsi, la reconstruction exacte du signal s'appuie sur des opérations duales d'arrondis (supérieurs ou inférieurs), appliquées lors des transformations directe et inverse entières. La transformée bidimensionnelle  $WHT_{2 \times 2}$  s'effectue par l'application successive de la transformée précédemment définie sur les lignes et sur les colonnes de l'image.

Nous introduisons dans cette section une nouvelle forme de la transformée bidimensionnelle  $WHT_{2 \times 2}$ , la  $RWHT_{2 \times 2}$  (Reversible  $WHT_{2 \times 2}$ ) 2D : la RWHT est alors directement applicable sur un bloc de taille  $2 \times 2$ .

### 1.1 Définition de la RWHT

Soit  $U_{2 \times 2}$  le bloc d'entrée, tel que

$$U_{2 \times 2} = \begin{bmatrix} u_0 & u_1 \\ u_2 & u_3 \end{bmatrix}. \quad (7.3)$$

Le bloc transformé  $Z_{2 \times 2}$  par la RWHT 2D est défini par

$$\begin{aligned} Z_{2 \times 2} &= WHT_{2 \times 2}(U_{2 \times 2}) = W_{2 \times 2} U_{2 \times 2} W_{2 \times 2} = \begin{bmatrix} z_0 & z_1 \\ z_2 & z_3 \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} u_0 + u_1 + u_2 + u_3 & u_0 + u_1 - u_2 - u_3 \\ u_0 - u_1 + u_2 - u_3 & u_0 - u_1 - u_2 + u_3 \end{bmatrix}. \end{aligned} \quad (7.4)$$

Les coefficients transformés sont par nature réels. Pour obtenir des valeurs entières, nous introduisons  $\hat{Z}_{2 \times 2}$  le bloc de valeurs entières, **arrondi** de  $Z_{2 \times 2}$ , tel que

$$\hat{Z}_{2 \times 2} = \text{Round}(Z_{2 \times 2}) = \begin{bmatrix} \hat{z}_0 & \hat{z}_1 \\ \hat{z}_2 & \hat{z}_3 \end{bmatrix} = \begin{bmatrix} \text{Round}_{z_0}[z_0] & \text{Round}_{z_1}[z_1] \\ \text{Round}_{z_2}[z_2] & \text{Round}_{z_3}[z_3] \end{bmatrix}. \quad (7.5)$$

$\text{Round}_{z_i}[\cdot]$  représente l'opérateur d'arrondi, qui peut a priori être soit un arrondi inférieur ( $\lfloor \cdot \rfloor$ ), soit supérieur ( $\lceil \cdot \rceil$ ).  $z_0$  correspond à la valeur continue du bloc *DC*, les autres coefficients  $z_i$  aux composantes fréquentielles *AC*.

La transformation inverse est identique à la transformation directe. Ainsi, la notation  $\tilde{U}_{2 \times 2}$  indique le **bloc transformé inverse** de  $\hat{Z}_{2 \times 2}$ , ce qui s'écrit :

$$\tilde{U}_{2 \times 2} = WHT(\hat{Z}_{2 \times 2}),$$

et  $\hat{U}_{2 \times 2}$  le bloc arrondi de  $\tilde{U}_{2 \times 2}$ .

La définition d'une transformation réversible implique  $\hat{U}_{2 \times 2} = U_{2 \times 2}$ , et ce malgré les opérations successives d'arrondis. Pour ce faire, nous avons défini une méthode originale de contrôle des arrondis, décrite dans la section suivante.

## 1.2 Opérations d'arrondis - Fonction de parité

Afin d'obtenir le caractère réversible de la transformation, une étape dédiée au contrôle des opérations d'arrondis est réalisée par l'exploitation d'une fonction de parité  $P(\cdot)$  :

$$P(x) = \begin{cases} o, & \text{si } x \text{ impair;} \\ e, & \text{si } x \text{ pair.} \end{cases}, \quad x \in \mathbb{N}. \quad (7.6)$$

En écrivant  $z_0$  sous la forme

$$z_0 = \lfloor z_0 \rfloor + \frac{\epsilon}{2}, \epsilon \in \{0, 1\},$$

puis par substitution dans l'équation (7.4),  $Z_{2 \times 2}$  peut être réécrit de telle sorte que

$$Z_{2 \times 2} = \frac{1}{2} \begin{bmatrix} 2 \lfloor z_0 \rfloor + \epsilon & 2 (\lfloor z_0 \rfloor - u_2 - u_3) + \epsilon \\ 2 (\lfloor z_0 \rfloor - u_1 - u_3) + \epsilon & 2 (\lfloor z_0 \rfloor - u_1 - u_2) + \epsilon \end{bmatrix}. \quad (7.7)$$

Cette expression montre clairement que les coefficients transformés seront soit tous entiers, soit tous non entiers, selon la valeur de  $\epsilon$ . Nous distinguons donc les deux cas possibles.

### 1.2.1 Somme paire des coefficients

Si  $P(\sum_{i=0}^3 u_i) = e$ , alors  $\epsilon = 0$  et  $\hat{Z}_{2 \times 2} = Z_{2 \times 2}$ . Cette précédente relation conduit à reconstruire des valeurs entières. Ainsi, on peut écrire

$$\tilde{u}_0 = \frac{1}{2} (4 \lfloor z_0 \rfloor - 2(u_1 + u_2 + u_3)) = \frac{1}{2} (2u_0) = u_0 \quad (7.8)$$

et par conséquent  $\hat{u}_0 = u_0$ .



### 1.2.2 Somme impaire des coefficients

Si  $P(\sum_{i=0}^3 u_i) = o$ , alors  $\epsilon = 1$ . Le problème de l'arrondi de  $Z_{2 \times 2}$  peut être ramené à celui de l'arrondi appliqué à la valeur  $\epsilon/2$  pour chaque coefficient.

Soit  $\Delta_i \in \{0, 1\}$  l'arrondi de  $\epsilon/2$  pour  $z_i$

$$\Delta_i = \text{Round}_{z_i} \left[ \frac{\epsilon}{2} \right] = \frac{\epsilon}{2} + \frac{\epsilon_i}{2}, \quad \epsilon_i \in \{-1, +1\}$$

Alors,  $\hat{Z}_{2 \times 2}$  s'exprime par

$$\hat{Z}_{2 \times 2} = \begin{bmatrix} \lfloor z_0 \rfloor + \Delta_0 & \lfloor z_0 \rfloor - u_2 - u_3 + \Delta_1 \\ \lfloor z_0 \rfloor - u_1 - u_3 + \Delta_2 & \lfloor z_0 \rfloor - u_1 - u_2 + \Delta_3 \end{bmatrix} = Z_{2 \times 2} + \frac{1}{2} \begin{bmatrix} \epsilon_0 & \epsilon_1 \\ \epsilon_2 & \epsilon_3 \end{bmatrix}. \quad (7.9)$$

Par conséquent, la matrice des coefficients reconstruits est donnée par l'expression suivante :

$$\tilde{U}_{2 \times 2} = \frac{1}{2} \begin{bmatrix} 2(u_0 - \epsilon) + (\Delta_0 + \Delta_1 + \Delta_2 + \Delta_3) & 2u_1 + (\Delta_0 + \Delta_1 - \Delta_2 - \Delta_3) \\ 2u_2 + (\Delta_0 - \Delta_1 + \Delta_2 - \Delta_3) & 2u_3 + (\Delta_0 - \Delta_1 - \Delta_2 + \Delta_3) \end{bmatrix} \quad (7.10)$$

De ce fait, une reconstruction exacte implique le respect du système d'équations

$$\begin{cases} \Delta_0 + \Delta_1 + \Delta_2 + \Delta_3 - 2\epsilon = 0 \\ \Delta_0 + \Delta_1 - \Delta_2 - \Delta_3 = 0 \\ \Delta_0 - \Delta_1 + \Delta_2 - \Delta_3 = 0 \\ \Delta_0 - \Delta_1 - \Delta_2 + \Delta_3 = 0 \end{cases} \quad \text{soit} \quad \begin{cases} \Delta_0 + \Delta_1 + \Delta_2 + \Delta_3 = 2\epsilon = 2 \\ \Delta_0 + \Delta_1 = \Delta_2 + \Delta_3 \\ \Delta_0 + \Delta_2 = \Delta_1 + \Delta_3 \\ \Delta_0 + \Delta_3 = \Delta_1 + \Delta_2 \end{cases} \quad (7.11)$$

De manière évidente, le système d'équations sur les valeurs de  $\Delta_i$  ne peut être résolu : aucune stratégie systématique d'arrondi ne peut prétendre aboutir à une reconstruction exacte.

La solution alternative vise à contrôler les valeurs d'arrondi de telle sorte que, lors de la transformation inverse, il soit possible de distinguer les valeurs reconstruites entières des valeurs non entières. Ainsi, après avoir déterminé la parité initiale de la somme des coefficients, il s'avère aisé de corriger le cas échéant les coefficients d'entrée de la transformée.

Pour distinguer le cas impair du cas pair, il suffit d'observer les valeurs des coefficients  $\tilde{u}_i$ . En effet, si l'ensemble  $\{\Delta_i\}$  est tel que  $P(\sum_{i=0}^3 \Delta_i) = o$ , alors les coefficients  $\tilde{u}_i$  ne sont représentés que

par des valeurs réelles. Posons  $\sum_{i=0}^3 \Delta_i = 1$ , nous obtenons alors

$$\begin{aligned} \Delta_0 + \Delta_1 + \Delta_2 + \Delta_3 = 1 &\Rightarrow 4\frac{\epsilon}{2} + \frac{\epsilon_0}{2} + \frac{\epsilon_1}{2} + \frac{\epsilon_2}{2} + \frac{\epsilon_3}{2} = 1 \\ &\Rightarrow \epsilon_0 + \epsilon_1 + \epsilon_2 + \epsilon_3 = -2. \end{aligned} \quad (7.12)$$

Par exemple, le choix de l'ensemble des valeurs  $\{\epsilon_0 = 1, \epsilon_1 = \epsilon_2 = \epsilon_3 = -1\}$  répond à la condition (7.12).

Finalement, la transformée inverse s'effectue en deux étapes :

1. on réalise le calcul  $\tilde{U}_{2 \times 2} = WHT(\hat{Z}_{2 \times 2})$ ,
2. si  $\tilde{u}_i$  est un réel, alors nous recalculons  $\tilde{U}_{2 \times 2}$  de telle sorte que

$$\tilde{U}_{2 \times 2} = WHT \left( \hat{Z}_{2 \times 2} - \frac{1}{2} \begin{bmatrix} \epsilon_0 & \epsilon_1 \\ \epsilon_2 & \epsilon_3 \end{bmatrix} \right). \quad (7.13)$$

Il est ensuite facile de vérifier l'égalité  $\tilde{U}_{2 \times 2} = U_{2 \times 2}$  dans tous les cas.

## 2 Pyramide RWHT

### 2.1 Construction indépendante de la grille

Afin de réaliser une compression sans perte des images, nous nous plaçons dans le cas où l'information globale (image  $LR$  basse résolution) et la texture sont encodées par le même processus de descente de la pyramide. Ainsi, la construction de la pyramide, indépendamment de la partition Quadtree, est déterminée par la relation de récurrence suivante :

$$\begin{cases} l = 0, & Y_0(i, j) = I(i, j); \\ l > 0, & Y_l(i, j) = \left\lfloor \frac{1}{4} \sum_{k=0}^1 \sum_{m=0}^1 Y_{l-1}(2x+k, 2y+m) \right\rfloor \end{cases} \quad (7.14)$$

avec  $0 \leq i \leq N_x^l$ ,  $0 \leq j \leq N_y^l$ , où  $N_x^l = N_x/l$  et  $N_y^l = N_y/l$ .

La décomposition *top-down* de la pyramide consiste évidemment en l'encodage du bloc  $Z_l(b^2(i, j))$ , transformé (par la RWHT) du bloc  $Y_l(b^2(i, j))$ . Des relations 7.4 et 7.14, il est facile d'en déduire que

$$Y_{l+1}(i, j) = \left\lfloor \frac{z_{0l}(2i, 2j)}{2} \right\rfloor,$$

et par conséquent

$$z_{0l}(2i, 2j) = 2 \times Y_{l+1}(i, j) + \epsilon, \quad \epsilon \in \{0, 1\}. \quad (7.15)$$

Ainsi, la composante DC de chaque bloc est reconstruite sans ambiguïté à partir du bloc situé au niveau supérieur de la pyramide, et d'un bit supplémentaire. Le codage de ce bit s'effectue séparément des autres coefficients. Nous vérifions au paragraphe suivant le comportement de cette nouvelle transformée lorsqu'elle est appliquée aux images naturelles.

### 2.2 Performances de la RWHT

Les paragraphes précédents ont contribué à la définition de la transformée RWHT : la technique présentée assure la réversibilité de la transformée  $WHT_{2 \times 2}$  2D classique. Le tableau 7.1 présente les entropies d'ordre zéro obtenues après transformation en S et application de la  $RWHT_{2 \times 2}$  décrite précédemment. Nous nous sommes placés dans le même contexte que celui décrit dans [SP96a], à savoir : six niveaux de décomposition du signal, et une entropie estimée séparément pour chaque coefficient.

Image	Entropie (bpp)		
	Raw	S	RWHT
Barbara2	7.51	<b>5.45</b>	5.47
Hotel	7.57	5.11	<b>5.09</b>
Lena	7.44	4.77	<b>4.75</b>
Gold	7.60	5.08	<b>5.06</b>
Peppers	7.57	4.89	<b>4.87</b>
us	4.84	3.65	<b>3.64</b>
tools	7.52	<b>5.95</b>	<b>5.95</b>
Moyenne	7.15	4.99	<b>4.97</b>

TAB. 7.1: Entropie du premier ordre (bit/pixels) de la méthode pyramidale RWHT, avec la transformée classique S+P.

Nous constatons le fait suivant : notre transformée RWHT s'avère en moyenne plus performante en termes d'entropie que la transformée en S. Par ailleurs, il est possible de mettre en œuvre simplement le regroupement des coefficients comme nous l'avons précédemment indiqué lors de la construction des algorithmes LAR-APP et Interleaved S+P. Ainsi, avec un partitionnement de l'image selon une partition  $QP^{[16...2]}$  et pour  $Th = 30$ , l'entropie chute, par exemple sur l'image "Lena", de 4.73 bpp à 4.59 bpp. Nous nous attarderons plus longuement au cours de ce chapitre (§3.3) sur l'influence de la décomposition en taille de blocs variable dans un schéma progressif sans perte à base de la  $RWHT_{2 \times 2}$ .

### 3 Pyramide RWHT et Prédiction

Les étapes de prédiction et d'interpolation sont des fonctions du domaine spatial relativement proches, mais suivent néanmoins des objectifs différents. Une interpolation globale est implantée dans notre schéma scalable de codage afin de récupérer une image à pleine résolution à partir de n'importe quel niveau  $l > 0$  de la pyramide décodée. Nous verrons en section suivante comment une interpolation locale se conçoit dans un contexte de décomposition Quadtree.

Nous proposons ici une technique unique efficace à la fois pour l'estimation des coefficients et pour l'interpolation des images produites lors du décodage de la pyramide.

Rappel :  $\check{Y}_l(b^2(i, j))$  et  $\tilde{Y}_l(b^2(i, j))$  signifient respectivement le bloc estimé et reconstruit du bloc  $Y_l(b^2(i, j))$ . Les indices  $i$  et  $j$  respectent le sens indiqué précédemment au chapitre 5, figure 5.6.

#### 3.1 Définition générale de la phase de prédiction

Le procédé d'estimation des valeurs de coefficients s'appuie sur le voisinage à la fois inter- et intra-niveaux. Le niveau supérieur fournit la moyenne du bloc  $2 \times 2$  au niveau courant : nous avons ici abordé le problème comme celui de la reconstruction des quatre points du bloc considéré autour de leur moyenne en fonction des gradients locaux.

$$\begin{array}{l|l}
 \text{Initialisation :} & \\
 \check{Y}_l(b^2(i, j)) & = \tilde{Y}_{l+1}(i, j), \forall (i, j) \in \tilde{Y}_{l+1} \\
 \text{Estimation :} & \\
 \check{Y}_l(b^2(i, j)) & = \bigcup_{k,m=0}^1 \check{Y}_l(2i+k, 2j+m), \text{ avec} \\
 \check{Y}_l(2i+k, 2j+m) & = \tilde{Y}_{l+1}(i, j) + \beta_m \left( V_v(2i+k, 2j+m) - \tilde{Y}_{l+1}(i, j) \right) \\
 & \quad + \beta_k \left( V_h(2i+k, 2j+m) - \tilde{Y}_{l+1}(i, j) \right)
 \end{array} \tag{7.16}$$

$V_h$  et  $V_v$  représentent les valeurs de voisinage horizontal et vertical du point courant.  $\beta_m$  et  $\beta_k$  sont les coefficients multiplicatifs appliqués sur chaque gradient local (voir §3.2) ( $k$  et  $m$  prennent leur valeur dans l'ensemble  $\{0, 1\}$ ). Deux modes différents du prédicteur vont se distinguer par la prise en compte d'un voisinage isotropique (mode linéaire) ou directionnel (mode non linéaire).

**Estimation linéaire.** Ce mode consiste à considérer les deux voisins les plus proches en 4-connexité au point courant, de telle sorte que

$$\begin{cases} V_v(2i + k, 2j + m) = \check{Y}_l(2i + k, 2j - 1 + 3m) \\ V_h(2i + k, 2j + m) = \check{Y}_l(2i - 1 + 3k, 2j + m). \end{cases} \quad (7.17)$$

**Estimation non linéaire.** Notre algorithme servant à la fois d'estimateur et d'interpolateur, il peut être judicieux d'éviter de trop lisser les contours contenus dans l'image. Pour ce faire, nous avons défini un mode plus complexe qui consiste à identifier localement les contours. Sept configurations possibles de frontières dans un bloc  $2 \times 2$  ont été établies (voir figure 7.1); la classification dans l'une d'entre elles s'effectue à partir de mesures de gradients locaux. L'attribution des voisinages est alors fonction de cette orientation. Par exemple, la configuration dite "Horizontal" induit les voisinages  $V_h = V_v = \check{Y}_l(2i - 1 + 3k, 2j + m)$ .

Vertical	Horizontal	Diag. +45° 1	Diag. +45° 2	Diag. -45° 1	Diag. -45° 2	Plat																												
<table><tr><td>X</td><td>O</td></tr><tr><td>X</td><td>O</td></tr></table>	X	O	X	O	<table><tr><td>X</td><td>X</td></tr><tr><td>O</td><td>O</td></tr></table>	X	X	O	O	<table><tr><td>X</td><td>O</td></tr><tr><td>O</td><td>O</td></tr></table>	X	O	O	O	<table><tr><td>X</td><td>X</td></tr><tr><td>X</td><td>O</td></tr></table>	X	X	X	O	<table><tr><td>X</td><td>O</td></tr><tr><td>X</td><td>X</td></tr></table>	X	O	X	X	<table><tr><td>O</td><td>O</td></tr><tr><td>X</td><td>O</td></tr></table>	O	O	X	O	<table><tr><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td></tr></table>	X	X	X	X
X	O																																	
X	O																																	
X	X																																	
O	O																																	
X	O																																	
O	O																																	
X	X																																	
X	O																																	
X	O																																	
X	X																																	
O	O																																	
X	O																																	
X	X																																	
X	X																																	

FIG. 7.1: Ensemble de configurations de blocs frontière considérés

### 3.2 Coefficients multiplicatifs $\beta_m$ et $\beta_k$

Sans aucune phase de quantification, les valeurs de  $V_v$  et  $V_h$  sont associées à l'une des trois situations suivantes :

- la valeur observée -dans l'espace causal- est une valeur exacte de reconstruction (le pixel à la position considérée du niveau courant a déjà été traité et encodé sans perte),
- la valeur correspond à une valeur moyenne de bloc (le pixel à la position considérée n'a pas encore subi de traitement),
- la valeur rencontrée est une valeur interpolée (position précédemment traitée par interpolation mais non encodée).

Dans ce dernier cas, il existe une inter-dépendance entre pixel courant et pixel voisin, dans le sens où la valeur de ce voisin a partiellement été calculée à partir du pixel courant. Cette constatation implique l'existence d'une relation entre les coefficients  $\beta$  pour deux positions adjacentes de deux blocs, afin d'établir la symétrie des gradients relatifs et de garantir une linéarité de reconstruction. Ainsi, si  $\beta_1$  correspond au coefficient appliqué sur la position estimée courante, et  $\beta_0$  le coefficient utilisé pour la valeur interpolée précédente, la symétrie, décrite par

$$\left( \check{Y}_l(2i - 1, 2j) - \tilde{Y}_{l+1}(i - 1, j) \right) = - \left( \check{Y}_l(2i, 2j) - \tilde{Y}_{l+1}(i, j) \right)$$

impose la relation suivante :

$$\beta_0 = \frac{\beta_1}{1 - \beta_1}, \quad \beta_1 \in [0, 0.5]. \quad (7.18)$$

Pour  $\beta_1 = 0.25$ , la pente est régulière entre les deux point interpolés. De plus, nos expérimentations ont montré que pour cette valeur  $\beta_1 = 0.25$ , l'algorithme fournit la meilleure prédiction.

La figure 7.2 montre une image interpolée à partir du niveau 3 de la pyramide (blocs  $8 \times 8$ ). Le tableau 7.2 fournit des résultats comparatifs de codage sans perte des images. Nous remarquons la supériorité de notre schéma sur S+P.



FIG. 7.2: Image intermédiaire (niveau  $l = 3$ ) sous-échantillonnée (a) et interpolée (b) - Décomposition globale (0.089 bpp)

Image	Entropie (bpp)			
	Raw	CALIC	S+P	RWHT+P
Barbara2	7.51	4.93	5.04	5.06
Hotel	7.57	4.57	4.97	4.83
Lena	7.44	4.33	4.33	4.30
Gold	7.60	4.65	4.73	4.73
Peppers	7.57	4.58	4.67	4.54
us	4.84	3.60	3.78	3.78
tools	7.52	5.53	5.73	5.71
Moyenne	7.15	<b>4.60</b>	4.75	<b>4.71</b>

TAB. 7.2: Entropie du premier ordre (bit/pixels) de la méthode RWHT+P (mise en œuvre de la phase de prédiction), CALIC et S+P.

### 3.3 Utilisation de la partition Quadtree

L'algorithme RWHT+P fait intervenir le même concept que celui utilisé pour le LAR-APP (chapitre 5) et l'Interleaved S+P (chapitre 6), à savoir : la décomposition pyramidale s'effectue selon deux descentes successives, contraintes par la partition Quadtree préalablement calculée. La méthode d'estimation présentée dans ce chapitre fait aussi office d'interpolation efficace. La figure 7.3 propose en exemple une image pleine résolution reconstruite à partir de l'image obtenue au niveau  $l = 1$  de la pyramide : une interpolation non linéaire est adoptée pour le traitement des plus petits blocs (blocs de taille  $2 \times 2$ ).



FIG. 7.3: Exemple d'image intermédiaire sous-échantillonnée (a) et interpolée (b), pour une décomposition non régulière - Première descente pyramide,  $l=1$  : 0.39 bpp.

Du point de vue de la compression sans perte, nous montrons dans le tableau 7.3 que la séparation des symboles, selon des lois a priori<sup>3</sup>, améliore très nettement les performances de l'algorithme (colonne "RWHT+P Qd"). L'entropie moyenne observée en termes d'entropie d'ordre zéro s'élève à 4.50 bits par pixels : par comparaison, le schéma de codage CALIC n'affiche que 4.75 bpp en moyenne, soit 0.15 bpp de plus que RWHT !

Image	Entropie (bpp)				
	Raw	CALIC	S+P	RWHT+P	RWHT+P Qd
Barbara2	7.51	4.93	5.04	5.06	<b>4.89</b>
Hotel	7.57	4.57	4.97	4.83	<b>4.60</b>
Lena	7.44	4.33	4.33	4.30	<b>4.19</b>
Gold	7.60	4.65	4.73	4.73	<b>4.66</b>
Peppers	7.57	4.58	4.67	4.54	<b>4.43</b>
us	4.84	3.60	3.78	3.78	<b>3.26</b>
tools	7.52	5.53	5.73	5.71	<b>5.50</b>
Moyenne	7.15	4.60	4.75	4.71	<b>4.50</b>

TAB. 7.3: Entropie du premier ordre (bit/pixels) de la méthode RWHT+P, avec partition Quadtree (Qd) ou non, CALIC et S+P.

## 4 Modélisation ou classification de contexte dans l'espace transformé

Nous avons déjà souligné au chapitre 5 §7 la contribution importante des techniques de modélisation de contexte dans l'efficacité des schémas de codage. Cette assertion se révèle particulièrement vrai pour une compression réalisée sans perte. Nous introduisons ici une méthode

<sup>3</sup>La séparation des lois de probabilité des symboles encodés s'avère identique à celle présentée en section 7 du chapitre 5.

originale de classification des symboles à transmettre, fondée sur une **mesure d'énergie**.

$\delta_{z_i}$  représente la valeur d'erreur commise sur un coefficient transformé lors du processus d'estimation, de telle sorte que  $\delta_{z_i} = z_i - \check{z}_i, 1 \leq i \leq 3$ . Il existe alors deux paramètres discriminants pour estimer la probabilité de l'erreur, à savoir :

- la dynamique de la valeur estimée  $\check{z}_i$ , traduisant une **estimation a priori de l'activité locale**,
- la différence entre la valeur moyenne du bloc  $z_0$  et la valeur estimée  $\check{z}_0$  correspondante, soit une **mesure objective de l'erreur commise** sur le bloc entier.

Clairement, ces deux paramètres ne sont pas indépendants, et nous avons donc défini une mesure d'énergie  $E_{z_i}$  associée à la valeur  $\delta_{z_i}$ , telle que

$$E_{z_i} = w_0 |\check{z}'_0| + w_1 |\check{z}_i|. \quad (7.19)$$

Les poids  $w_0$  et  $w_1$  ont été fixés empiriquement et respectivement à 0.5 et 1.

Classiquement, une quantification est alors opérée, afin d'éviter toute dilution de contexte. Nous avons pour notre part opté pour une technique plus souple de classification de contexte inter-coefficients, dépendante de  $E_{z_i}$ . La fonction de classification a été établie de manière à ce que pour chaque parcours des coefficients  $\delta_{z_i}$ , **les populations soient identiques dans chacune des classes produites**.

Soit  $Nb_{class}$  le nombre maximal de classes autorisées. On note  $p_{E_z}$  la probabilité suivante :  $p_{E_z} = p(E_z = E_{z_i})$ . Nous introduisons de plus le paramètre  $d_n$ , avec  $d_0 = \infty$  et  $d_{Nb_{class}} = 0$ . Il est alors possible d'écrire

$$\forall d_n \in \{d_1 \dots d_{Nb_{class}}\}, \quad \int_{E_z=d_{n-1}}^{d_n} p_{E_z} = \frac{1}{Nb_{class}} \quad (7.20)$$

Les valeurs  $d_n$  constituent les seuils de décision des classes, et doivent être transmises au décodeur.

Une classe  $C_n$  se calcule alors comme suit :

$$C_n = \{E_z | d_{n-1} > E_z \geq d_n\}, \quad (7.21)$$

avec  $1 \leq n \leq Nb_{class}$ .

L'appartenance aux différentes classes peut se juger comme une mesure relative de l'activité entre coefficients. La classe  $C_1$  correspond aux coefficients à activité maximale. Nous avons observé le fait suivant : au delà de 4 classes, la classification n'apporte plus de gain de compression. En pratique, le nombre de classes maximal utile s'avère égal à 4 pour le niveau 0, 3 pour le 1, etc. On a également pu constater que, compte tenu de la faible activité des coefficients  $\delta_{z_3}$ , pour un même niveau de pyramide, le nombre de classes utiles est globalement égal à la moitié de celui utilisé pour les autres coefficients. Nous avons donc fixé pour,  $\delta_{z_3}$ , le nombre de classes à  $\left\lfloor \frac{Nb_{class}}{2} \right\rfloor$ .

Les performances de notre méthode de classification sont indiquées dans le tableau 7.4 (colonne "RWHT+P Cl"). Nous remarquons notamment que le partitionnement et la classification de contexte apportent individuellement et en moyenne la même contribution. Il est à noter que, en fonction des images, l'un peut se montrer plus efficace que l'autre.

Image	Entropie (bpp)					
	Raw	CALIC	S+P	RWHT+P	RWHT+P Qd	RWHT+P Cl
Barbara2	7.51	4.93	5.04	5.06	<b>4.89</b>	4.90
Hotel	7.57	4.57	4.97	4.83	<b>4.60</b>	4.62
Lena	7.44	4.33	4.33	4.30	4.19	<b>4.16</b>
Gold	7.60	4.65	4.73	4.73	4.66	<b>4.64</b>
Peppers	7.57	4.58	4.67	4.54	4.43	<b>4.41</b>
us	4.84	3.60	3.78	3.78	3.26	<b>3.25</b>
tools	7.52	5.53	5.73	5.71	5.50	<b>5.48</b>
Moyenne	7.15	4.60	4.75	4.71	4.50	<b>4.49</b>

TAB. 7.4: Entropie du premier ordre (bit/pixels) de la méthode RWHT+P, avec partition Quadtree (Qd) ou non, classification (Cl) ou non, CALIC et S+P.

## 5 Combinaison des deux approches - Résultats

La combinaison de la décomposition Quadtree et de la modélisation de contexte conduit à une solution optimale (tableau 7.5), **largement supérieure** aux méthodes de l'état de l'art S+P et CALIC (colonne "RWHT+P QD & Cl"), de l'ordre de **0.17 bpp en moyenne!** En effet, les deux optimisations conjuguées aboutissent en général à améliorer les résultats. Une seule image fait exception ("Zelda" qui se trouve être aussi l'image la plus simple), et où les lois de distribution des erreurs de prédiction restent les plus homogènes.

De plus, quelle que soit la complexité des images naturelles (de "Baboon" à "Zelda"), la supériorité de notre approche sur CALIC se chiffre entre 0.11 et 0.19 bpp. CALIC reste en tête pour une seule image ("Finger") dont les caractéristiques statistiques et visuelles ne peuvent réellement la qualifier d'image naturelle. C'est aussi la seule image pour laquelle ni le partitionnement (découpage uniquement en petits blocs du fait de la grande dynamique locale sur toute l'image), ni la classification (très faible capacité de prédiction a priori) ne contribuent à améliorer significativement le codage. Même dans ce cas très particulier, les résultats (5.54 bpp) se situent néanmoins pratiquement au niveau de ceux de CALIC (5.52 bpp).

Enfin, notre technique demeure bien sûr multirésolution et "scalable". Pour les six niveaux de décomposition considérés, l'encodage total d'une image comporte onze flux ( $1 + 2 \times 5$ ) successifs. La figure 7.4 illustre le type de reconstruction progressive possible, en fournissant quelques unes des images intermédiaires.

Les résultats de codage surpassent évidemment les techniques progressives de codage sans perte basées ondelettes qui se trouvent être au deçà de ceux de CALIC. Le succès de notre méthode est sans doute lié à la capacité de conjuguer une décomposition pyramidale non redondante de type transformation en ondelettes réversible, avec une phase de prédiction efficace analogue à ce que l'on peut généralement observer pour les pyramides spatiales.



Image	Entropie (bpp)						
	Raw	CALIC	S+P	RWHT+P	RWHT+P Qd	RWHT+P Cl	RWHT+P Qd&Cl
Baboon	7.36	6.14	6.10	6.11	6.00	6.01	<b>5.95</b>
Barbara2	7.51	4.93	5.04	5.06	4.89	4.90	<b>4.85</b>
Hotel	7.57	4.57	4.97	4.83	4.60	4.62	<b>4.54</b>
Lena	7.44	4.33	4.33	4.30	4.19	4.16	<b>4.14</b>
Gold	7.60	4.65	4.73	4.73	4.66	4.64	<b>4.61</b>
Finger	7.41	<b>5.52</b>	5.57	5.57	5.56	5.55	5.54
Peppers	7.57	4.58	4.67	4.54	4.43	4.41	<b>4.38</b>
us	4.84	3.60	3.78	3.78	3.26	3.25	<b>3.09</b>
tools	7.52	5.53	5.73	5.71	5.50	5.48	<b>5.41</b>
Zelda	7.27	3.98	3.95	3.91	3.90	<b>3.84</b>	3.87
Moyenne	7.21	4.78	4.89	4.85	4.70	4.69	<b>4.63</b>

TAB. 7.5: Entropie du premier ordre (bit/pixels) de la méthode RWHT+P, avec partition Quadtree (Qd) ou non, classification (Cl) ou non, CALIC et S+P.



a. Première descente de la pyramide -  $l_{min} = 5$  (1 flux) : 0.005 bpp



b. Première descente de la pyramide -  $l_{min} = 3$  (3 flux) : 0.073 bpp



c. Première descente de la pyramide -  $l_{min} = 1$  (5 flux) : 0.433 bpp



d. Première descente de la pyramide -  $l_{min} = 0$  (6 flux) : 1.194 bpp



e. Deuxième descente de la pyramide -  $l_{min} = 1$  (10 flux) : 2.014 bpp



f. Deuxième descente de la pyramide -  $l_{min} = 0$  (11 flux) : 3.87 bpp

FIG. 7.4: Codage progressif sans perte sur l'image "Zelda"

## 6 Conclusion

Nous avons dans ce chapitre décrit une nouvelle méthode pyramidale de codage conjoint avec et sans perte. Partant d'une technique a priori bien connue, la *WHT* a été réécrite et conduit à une solution particulièrement efficace. En particulier, la définition d'une version réversible de la *WHT* classique constitue une étape importante quant à la possibilité d'envisager une méthode commune efficace allant des codages très bas débit jusqu'au sans perte. Une décomposition pyramidale basée sur une  $WHT_{2 \times 2}$  a ensuite été élaborée.

Une phase commune de filtrage/prédiction sert aussi bien de post-traitement que de prédicteur dans l'espace transformé. Un effort important a également été porté sur la façon de coder les coefficients transformés. L'élaboration d'une stratégie de classification de contexte a notamment permis l'obtention de lois entropiques plus homogènes. Cette stratégie procure non seulement un gain en compression, mais aussi en termes de progressivité.

Les résultats obtenus en compression sans perte se situent **largement au delà des performances des méthodes de l'état de l'art**. La méthode RWHT+P se révèle supérieure encore à notre algorithme Interleaved S+P. Contrairement à ce dernier, l'étude du RWHT+P s'est cantonnée à l'observation de son comportement sur les images dites naturelles. Au vu des résultats obtenus par l'Interleaved S+P, nous nous devons d'analyser également l'application de l'algorithme RWHT+P sur les images médicales en particulier, afin de construire la meilleure méthode possible. Par ailleurs, étant donnée la contribution de la classification de contexte sur l'entropie globale résultante, il semble judicieux d'envisager l'implantation de cette solution pour l'Interleaved S+P.

Enfin, d'une façon générale, il serait intéressant de coupler la méthode de codage intégrée à JPEG-2000 et nos techniques pyramidales afin de rendre réellement compte des performances de nos solutions par comparaison avec ce standard. Toutefois, certaines précautions de développement devront être prises, de sorte à conserver les propriétés innovantes des méthodes LAR. En particulier, le *zerotree* exploité par JPEG-2000 devra rester confiné au partitionnement donné par le codeur spatial du LAR.



Troisième partie

**Le LAR en télé médecine et sur  
systèmes embarqués**



## Chapitre 8

# Télémédecine : transmission prioritaire et sécurisée d'images compressées par le LAR

La télémédecine, terme générique signifiant "médecine pratiquée à distance", constitue désormais une discipline porteuse, tant du point de vue recherche que du point de vue industriel. L'objectif de l'approche est d'employer les technologies disponibles pour transférer les données cliniques et les compétences médicales, plutôt que de déplacer les patients ou les praticiens. En particulier, la mise en œuvre généralisée du PACS<sup>1</sup>, système d'archivage et de communication des images médicales, dans toutes les unités hospitalières en France, nécessite un choix technologique relatif aux besoins de compression et de sécurisation des images.

Dans ce contexte, une structure de réflexions regroupant praticiens et chercheurs a vu le jour au sein du très récent GdR STIC Santé. Les discussions portent non seulement sur les fonctionnalités souhaitables d'un schéma de compression (multirésolution, rapidité d'accès aux différents niveaux de résolution, codage par Régions d'Intérêt, images basse résolution de qualité), mais aussi sur l'aspect sécurisation de l'image et/ou de sa transmission sur réseaux hauts/bas débits.

Nous proposons ici une solution de codage conjoint source-canal, par l'implantation successive d'une approche pyramidale du LAR et de la transformée Mojette. Cette méthode a donné lieu à de nombreuses communications et publications, à partir desquelles ce chapitre s'est construit [BDR04a, DBR<sup>+</sup>04, BDR<sup>+</sup>04b, BPD<sup>+</sup>05]. Ces travaux se sont ainsi effectués en étroite collaboration avec les "pères" de la Mojette, à savoir l'équipe Image et Video Communication (IVC) de l'IRCCyN à Nantes<sup>2</sup>.

La section 1 définit le contexte général de cette étude, à travers la présentation du PACS. Puis, au paragraphe 2, sont posées les hypothèses sur lesquelles notre étude d'un codage conjoint s'appuie. Afin de mieux cibler l'intérêt de la protection hiérarchique adaptative par la Mojette, nous cherchons aussi à comprendre l'impact d'une erreur binaire opérée sur le flux transmis (section 3). La transformée Mojette est ensuite explicitée (§4). Enfin, nous donnons la recette du LAR aux Mojettes, utilisable dans le domaine de la télémédecine, pour un besoin de transmission sécurisée d'images compressées. Une étude de l'apport de la technique conjointe conclue ce chapitre.

---

<sup>1</sup>PACS : Picture Archiving and Communication System

<sup>2</sup><http://penalty@M://www.irccyn.ec-nantes.fr>

## 1 Contexte de l'étude : la télémédecine - Analyse corrélée au PACS

L'élimination du support physique traditionnel (film radiologique, papier...) par des techniques d'imagerie du type IRM ou encore scanner, a considérablement fait évoluer les métiers à caractère médical. En effet, la numérisation des données a pour vocation d'accélérer les traitements, la délivrance d'un diagnostic, et d'affiner l'observation d'une pathologie donnée par la comparaison avec des résultats antérieurs.

La télémédecine regroupe plusieurs formes différentes de communications entre centres de soins distants. Nous pouvons citer à cet effet la téléconsultation (consultation interactive du dossier patient à distance), et le télédiagnostic (diagnostic primaire élaboré à distance, par le spécialiste non présent dans l'unité hospitalière). La téléradiologie est une sous-classe naturelle du télédiagnostic [FC00]. Dans ce cadre, il a fallu définir des règles de communications et de stockage des images médicales produites. Une définition du PACS pourrait être la suivante [Hoc03] :

**Définition 8.1 (PACS - Picture Archiving and Communication Systems)** *Le PACS est un système intégré de gestion électronique des images médicales avec des fonctions d'archivage et de communication. Il rend possible le cycle suivant : acquisition des images, archivage électronique, impression, communication et consultation via un réseau, visualisation, traitement et interprétation sur stations dédiées.*

Il est à noter que le PACS repose sur le standard DICOM de communication et de description des images<sup>3</sup>. Une synthèse des principes du PACS, ainsi que la description de plusieurs systèmes opérationnels sont disponibles grâce aux travaux de Bernard Gibaud (Université de Rennes 1 - UFR Faculté de Médecine) [Gib00]. Désormais le PACS ne peut plus se concevoir sans évoquer d'une part le SIH (Système d'Information Hospitalier), avec lequel il communique toute demande d'examens et résultats associés, et d'autre part le SIR (Système d'Information Radiologique), permettant la gestion d'un service de radiologie complet. Les praticiens du service de radiologie de l'Hôpital Européen Georges Pompidou (Paris)<sup>4</sup> préfèrent, dans ce cadre, évoquer les fonctionnalités de l'I4 (Image, Information, Informatique, Intégration) [FHH04, FHR<sup>+</sup>04].

La figure 8.1 illustre ainsi le principe d'un système complet de téléradiologie, où apparaît le PACS et les relations entre les différents services et le personnel médical impliqués. Un examen réalisé sur un équipement d'imagerie médicale d'une unité radiologique est (après éventuelle numérisation) mis au format DICOM avant d'être stocké dans l'une des bases de données dédiées. Le radiologue, le spécialiste ou autre (urgentiste, généraliste...) a alors accès à l'image et procède à sa consultation. Via un serveur web sécurisé, les données peuvent être partagées avec d'autres centres de soins pour une consultation ultérieure ou encore pour l'établissement d'un diagnostic complémentaire.

<sup>3</sup>L'ensemble des documents concernant le standard DICOM peut être téléchargé à partir du site [http\unskip\penalty\@M://medical.nema.org/](http://medical.nema.org/)

<sup>4</sup><http\penalty\@M://www.radiologie-pompidou.org/>

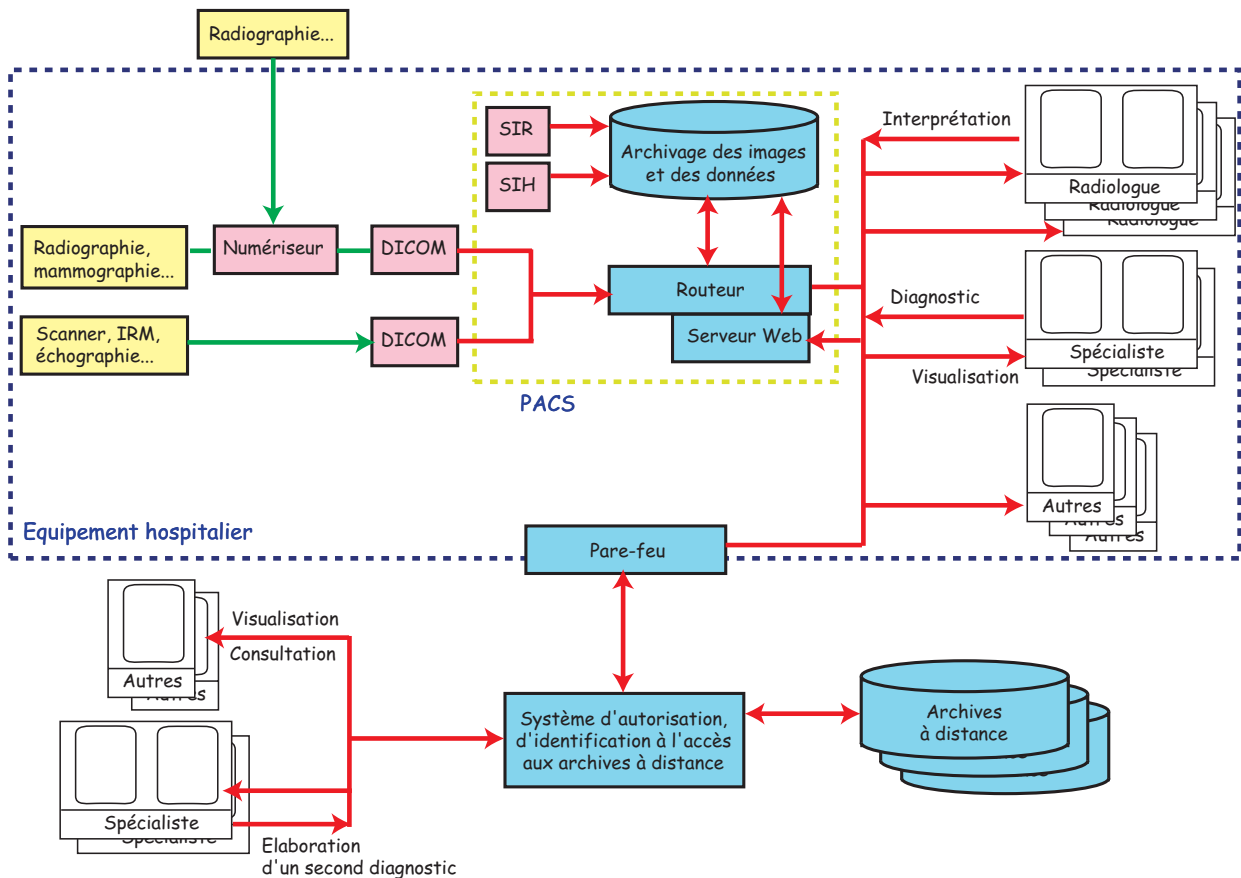


FIG. 8.1: Visualisation des différents services réalisés par le PACS, ainsi que son intégration dans un système de soin complet.

L'archivage s'effectue selon trois étapes successives : dans un premier temps, l'image est sauvegardée sur un serveur local, typiquement sur disques durs RAID <sup>5</sup> (accès immédiat, stockage à court terme), avant d'être transférée dans un juke-box de bandes magnétiques DLT <sup>6</sup>, pour une durée de 5 ans. La législation française prévoyant des durées de stockage minimales de 20 ans (plus, pour un individu mineur, le nombre d'années avant l'acquisition de sa majorité) à partir de la date d'examen, d'autres systèmes de stockage indirect doivent être mis en place (archivage sur étagères par Disques amovibles, CD-Rom, DVD-Rom) [Boz03].

Si la télémédecine est bien évidemment omniprésente dans un environnement hospitalier, elle se décline aussi dans le domaine des urgences [Pet04]. Cependant, l'imagerie médicale, n'est pas encore implantée dans les habitudes des urgentistes français (SAMU, SMUR). Suite à une table ronde lors du colloque "Les Nouvelles Technologies dans la Cité" <sup>7</sup>, il apparaît que, en France, la communauté médicale émet encore des réserves quant à ce type d'application. De nombreux pays procèdent à des essais de mise en œuvre de systèmes à visiophonie : en particulier, à Chypre, l'implication des politiques dans le projet de téléradiologie d'urgence a permis l'élaboration de structures de soins mobiles efficaces [AKP<sup>+</sup>04].

<sup>5</sup>La technologie RAID (acronyme de Redundant Array of Inexpensive Disks, soit Ensemble Redondant de Disques Indépendants) permet de constituer une unité de stockage à partir de plusieurs disques durs.

<sup>6</sup>Digital Linear Tape

<sup>7</sup><http://penalty@M:/tic-cite.univ-rennes1.fr>



### Et la transmission ?

La figure 8.1 fait mention en rouge des voies de communication nécessaires à la télémédecine. Les réseaux font classiquement appel à des technologies filaires, et utilisent des liaisons sécurisées dédiées. De nombreux travaux visent à concevoir l'architecture de ces réseaux complexes inclus dans un système PACS [TKO96, TO01]. L'apport des nouvelles technologies liées au développement des réseaux sans fil permet d'imaginer des solutions avancées de télémédecine [PS04, PKV<sup>+</sup>02, Vuc03]. En particulier, des produits basés sur l'utilisation conjointe d'un PDA et d'une liaison sans fil existent déjà dans le domaine de la téléradiologie [NSHV04, Gre05].

Du point de vue de la sécurisation de la transmission pour de la téléradiologie, plusieurs points sont à satisfaire, à savoir [Med97] :

- accès limité aux ressources du système,
- accès limité aux données patients (authentification),
- garantie de l'intégrité des données,
- protection contre le piratage,
- confidentialité.

Ainsi, les composantes principales du mécanisme de sécurité liés au partage des images médicales sont l'encryptage, la présence de signature numérique, et de codes correcteurs efficaces.

## 2 Codage conjoint source-canal efficace : les hypothèses

Dans le domaine des systèmes de communication, les usagers des réseaux de type Internet s'attendent à une qualité de service <sup>8</sup> spécifiée de bout-en-bout. La transformée Mojette a été introduite dans le cadre des systèmes multimédias distribués sur Internet par Benoît Parrein [Par01]. Si la transformée Mojette peut s'insérer dans de nombreuses applications visant à la sécurité liée aux images (tatouage, cryptographie en particulier) [Aut02], nous traitons ici de la transmission de paquets IP sur les réseaux bas débits. Dans ce contexte, nous cherchons à résoudre ce problème récurrent : comment gérer au mieux les paquets IP perdus et donc non reçus ?

Pour ce faire, nous nous plaçons dans l'hypothèse de "temps mou" : aucune retransmission de l'information perdue n'est alors autorisée. De plus, on suppose l'intégrité des paquets produits, *i.e.* les couches MAC et PHY du protocole d'acheminement des paquets sont considérées comme efficaces.

Par ailleurs, à des fins de transmission de masses de données sur réseaux à bas-débits, il devient impératif de sélectionner une source scalable, afin de suppléer au mieux aux limitations dues aux réseaux. Les approches multirésolution du LAR conviennent évidemment à l'élaboration de solutions pour la télémédecine. Le codage source, déjà défini dans les chapitres précédents, fait à la section suivante l'objet d'une étude quant à sa robustesse aux erreurs de transmission.

---

<sup>8</sup>Qualité de service : QoS (Quality of Service)

### 3 Etude de l'impact d'une erreur binaire sur un flux LAR

Afin d'évaluer la robustesse de notre méthode, et de comprendre l'importance d'une valeur transmise par rapport à une autre, nous avons voulu tester l'impact visuel de différents types d'erreurs commises sur un flux LAR donné. Cette étude a été réalisée au moyen du codeur Interleaved S+P, aucun codeur entropique n'étant implanté. Cette précision est importante : nous ne sommes pour l'instant pas en mesure de réaliser l'étude codeur entropique compris. Dans cette section, nous examinons successivement l'effet d'une erreur réalisée sur les valeurs codant pour la grille, puis celle réalisée sur une valeur codant pour une erreur de prédiction.

#### 3.1 Erreur commise sur la grille

##### 3.1.1 Système d'encodage de la grille

Pour comprendre les répercussions d'une erreur commise sur une valeur codant pour la grille, il s'avère nécessaire de réfléchir à l'algorithme de codage de celle-ci. De nombreux travaux font état de l'encodage des partitionnements Quadtree : la technique la plus connue s'appuie sur la structure d'arbre régulière inhérente à cette représentation [Nic92]. En considérant  $k_{max} = \log_2(N_{max})$  et  $k_{min} = \log_2(N_{min})$ , le nombre de niveaux de la décomposition est donc égal à  $(k_{max} - k_{min} + 1)$ . L'algorithme d'encodage progressif de la structure Quadtree consiste alors à réaliser un codage binaire itératif conditionnel inter-niveaux, tel qu'un bloc non présent à la partition de niveau  $N$  se retrouve sub-divisé en quatre blocs au niveau  $N/2$ .

Nous proposons ici une méthode d'encodage de la grille améliorant sensiblement les performances entropiques. Si  $N_{max}$  et  $N_{min}$  correspondent respectivement à la taille maximale et à la taille minimale des blocs dans la partition Quadtree construite, l'encodage de la grille suppose  $\log_2(N_{max}) - \log_2(N_{min})$  étapes successives.

L'image des tailles notée  $Siz$  donne en tout point la taille des blocs telle que  $Siz(x, y) \in \{N_{min}, \dots, N_{max}\}$ . La première étape ( $i = 1$ ) de l'algorithme consiste en l'entière récupération de la taille des blocs positionnés sur les pixels de coordonnées  $p(x, y)$ , tels que

$$x \bmod N_{max} = 0 \text{ et } y \bmod N_{max} = 0.$$

Les blocs rencontrés peuvent être de taille  $N_{min} \times N_{min}, \dots, N_{max} \times N_{max}$ , et sont donc encodés respectivement par les valeurs  $0, 1, \dots, \log_2(N_{max}) - \log_2(N_{min})$ .

Pour les étapes  $i = 2, \dots, \log_2(N_{max}) - \log_2(N_{min})$  suivantes, l'algorithme procède comme suit. Nous considérons les pixels de coordonnées  $p(x, y)$  tels que

$$\begin{cases} x \bmod \left(\frac{N_{max}}{2^{i-1}}\right) = 0, & y \bmod \left(\frac{N_{max}}{2^{i-1}}\right) = 0 \\ x \bmod \left(\frac{N_{max}}{2^{i-2}}\right) \neq 0, & y \bmod \left(\frac{N_{max}}{2^{i-2}}\right) \neq 0 \end{cases}$$

Si  $Siz(x, y) \leq \frac{N_{max}}{2^{i-1}}$ , une donnée est alors transmise et prend la valeur  $\log_2(Siz(x, y)) - \log_2(N_{min})$ .

Certains blocs restent non codés lors du parcours de la grille par les différentes étapes présentées ci-dessus : il n'est en effet pas nécessaire de préciser leur taille, forcément égale à  $N_{min} \times N_{min}$ . La figure 8.2 illustre la méthode de codage pour un pavage classique  $QP^{[16...2]}$ .

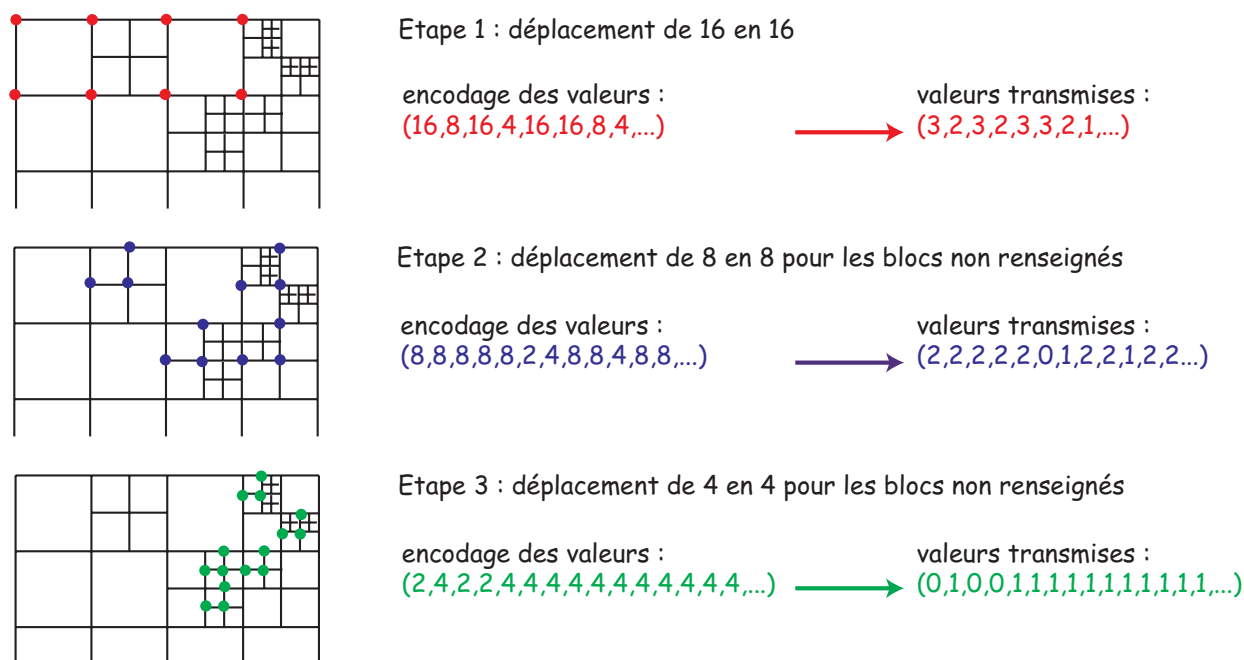
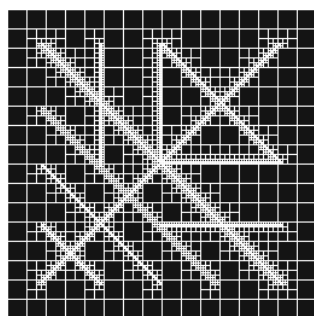


FIG. 8.2: Les 3 étapes d'encodage de la grille valant pour une partition Quadtree autorisant les blocs de taille  $2 \times 2$  à  $16 \times 16$ .



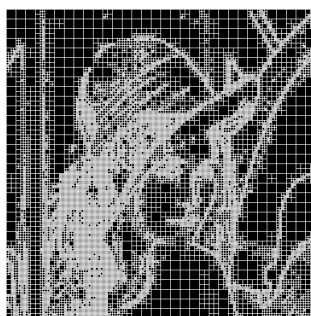
A1. crosses : image originale



A2. crosses : grille associée



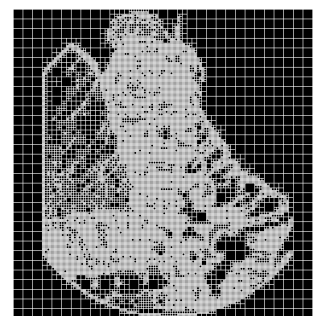
B1. lena : image originale



B2. lena : grille associée



C1. colon : image originale



C2. colon : grille associée

FIG. 8.3: Images utilisées et grilles originales associées

### 3.1.2 Erreur et effet visuel

Trois images ont principalement été testées, à savoir : "crosses", "lena" et "colon". Les trois images sont données figure 8.3, ainsi que leur représentation non uniforme associée obtenue au codeur (représentation originale sans insertion d'erreur). L'image "crosses" permet de visualiser et analyser au mieux les déformations induites dans des zones complètement uniformes et dans les zones à très forte activité locale. Pour les images naturelles classiques, "lena" est - quasiment - incontournable, et les effets visuels peuvent être, selon certains spécialistes, extrêmement dommageables à l'appréciation de la qualité produite. Enfin, l'image "colon" est un exemple issu du domaine médical, contexte qui nous préoccupe plus particulièrement dans ce chapitre.

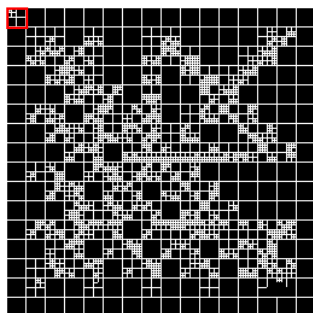
Afin d'évaluer l'impact d'une erreur commise sur la grille, nous modifions une unique valeur au décodeur, à la réception du flux. Nous avons adopté un pavage Quadtree  $QP^{[16...2]}$ , de telle sorte que trois parcours successifs (donc trois flux) suffisent à l'encodage de la grille. Commettre une erreur sur le premier flux risque bien évidemment de perturber d'autant plus la reconstruction de l'image finale qu'une erreur située dans l'un des deux autres flux. Un échantillon représentatif des observations que nous pouvons effectuer est donné dans les figures 8.4, 8.5, 8.6 et 8.7. Nous exposons les dégradations visuelles engendrées par le remplacement d'un bloc d'une taille donnée par un bloc de taille différente, et ce pour les trois niveaux d'encodage de la grille.

L'obtention d'une valeur erronée entraîne une modification de la représentation non uniforme de l'image. Typiquement, si le premier bloc en haut à gauche de l'image est originellement de taille  $n \times n$ , contraindre le bloc à la taille  $m \times m$  modifie le découpage ultérieur de l'ensemble des blocs concernés par la deuxième étape de l'algorithme. De plus, le mode de parcours de la grille s'effectuant ligne à ligne, seule la partie **anti-causale** de l'image est affectée par l'erreur produite : les conséquences visuelles sont donc plus ou moins dramatiques en fonction de la position de cette erreur dans le flux affecté (voir figure 8.5.A2).

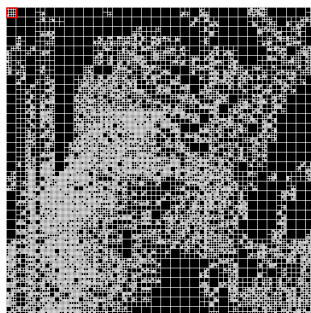
Les exemples présentés ici montrent clairement l'impossibilité de reconstruire correctement toute image au décodeur en cas de mauvaise réception de l'information de grille. Naturellement, la qualité visuelle de l'image reconstruite augmente si l'erreur n'est commise que sur les étapes ultérieures de décodage (figures 8.6 et 8.7). La valeur du PSNR est indiquée ici de façon à quantifier la dégradation relative entre deux images reconstruites successives.

### 3.1.3 Interprétations et applications possibles

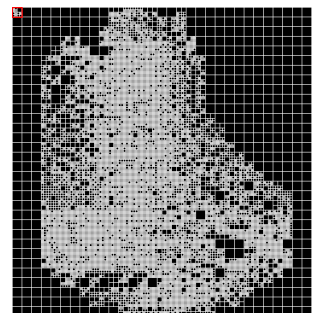
Une première et simple constatation est la suivante : l'information codant pour la grille est absolument essentielle à toute reconstruction correcte de l'image. Ainsi, dans le cadre de la protection des paquets IP pour une transmission bas-débits, une **protection maximale** doit être assignée aux sous-flux codant la représentation non uniforme. Cette propriété peut aussi être entretenue à des fins de **cryptographie** : seul le destinataire connaissant les méthodes de rétablissement de la grille originelle est capable de reconstruire parfaitement l'image transmise.



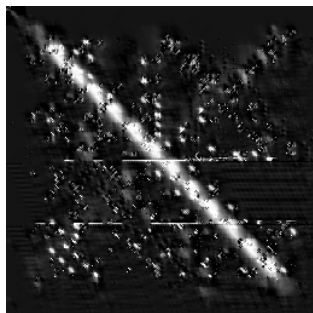
A1. crosses : grille modifiée



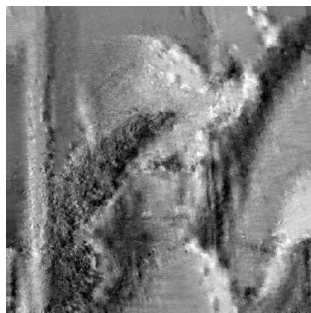
B1. lena : grille modifiée



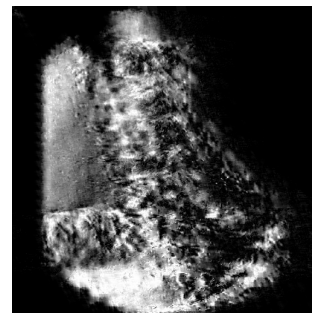
C1. colon : grille modifiée



A2. crosses : image reconstruite,  $PSNR = 13.58$  dB

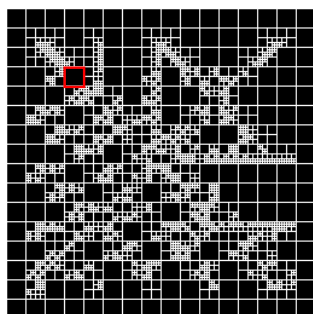


B2. lena : image reconstruite,  $PSNR = 5.22$  dB

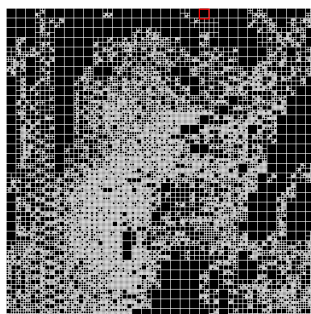


C2. colon : image reconstruite,  $PSNR = 14.99$  dB

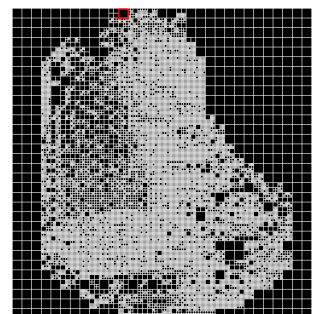
FIG. 8.4: Première étape du codage de grille : remplacement d'un bloc  $16 \times 16$  par un bloc  $2 \times 2$ .



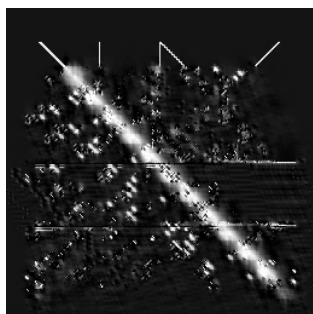
A1. crosses : grille modifiée



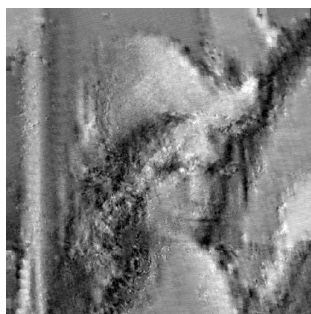
B1. lena : grille modifiée



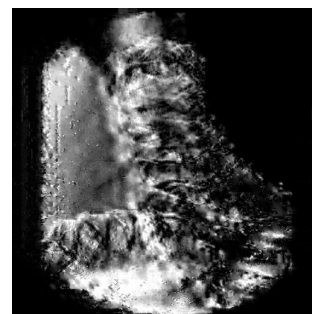
C1. colon : grille modifiée



A2. crosses : image reconstruite,  $PSNR = 13.88$  dB



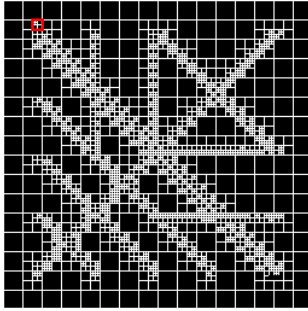
B2. lena : image reconstruite,  $PSNR = 18.47$  dB



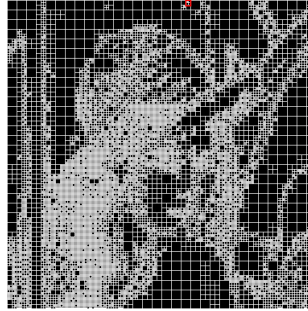
C2. colon : image reconstruite,  $PSNR = 15.19$  dB

FIG. 8.5: Première étape du codage de grille : remplacement d'un bloc  $2 \times 2$ ,  $4 \times 4$  ou  $8 \times 8$  par un bloc  $16 \times 16$  d'un bloc plus petit.

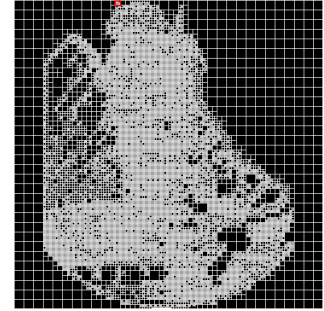




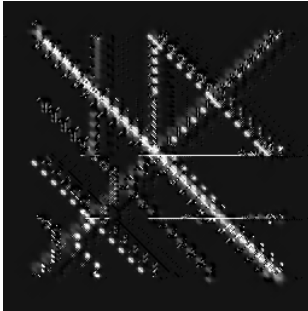
A1. crosses : grille modifiée



B1. lena : grille modifiée



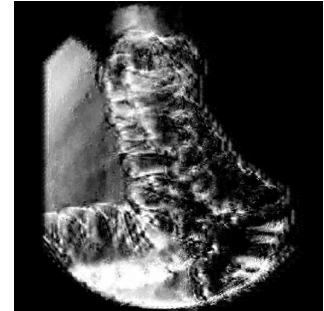
C1. colon : grille modifiée



A2. crosses : image reconstruite,  $PSNR = 15.28$  dB

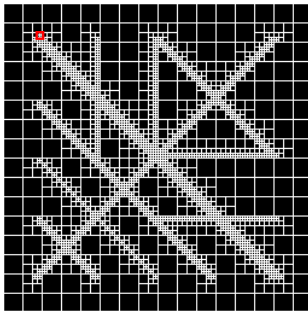


B2. lena : image reconstruite,  $PSNR = 21.60$  dB

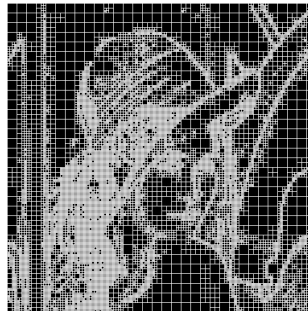


C2. colon : image reconstruite,  $PSNR = 17.55$  dB

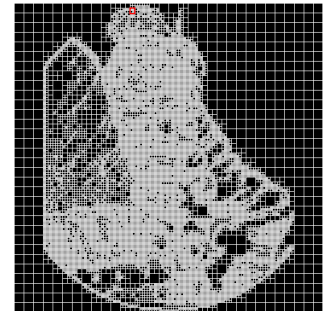
FIG. 8.6: Deuxième étape du codage de grille : remplacement d'un bloc  $8 \times 8$  par un bloc  $2 \times 2$ .



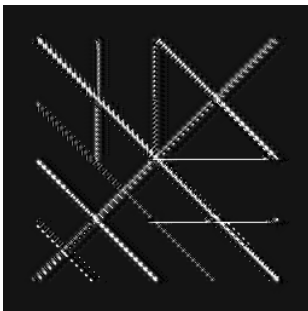
A1. crosses : grille modifiée



B1. lena : grille modifiée



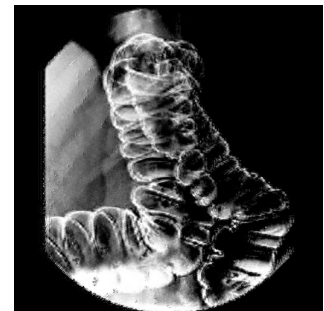
C1. colon : grille modifiée



A2. crosses : image reconstruite,  $PSNR = 17.42$  dB



B2. lena : image reconstruite,  $PSNR = 25.52$  dB



C2. colon : image reconstruite,  $PSNR = 21.77$  dB

FIG. 8.7: Troisième étape du codage de grille : remplacement d'un bloc  $2 \times 2$  par un bloc  $4 \times 4$ .

Une question importante doit être soulevée : *dans quelle mesure peut-on connaître le contenu de l'image si la grille est une information inconnue, ou si un processus perturbe la récupération correcte de cette grille ?*

Pour y répondre, deux méthodes sont envisageables. La première consiste à essayer toutes les combinaisons de blocs possibles, et à comparer les images produites, en espérant retrouver l'image telle qu'elle a été transmise. Raisonnons sur l'élaboration de la première étape d'encodage de cette grille. Sans information a priori, chaque valeur de bloc possède la même probabilité d'apparition. Pour un bloc,  $\log_2 \frac{N_{max}}{N_{min}} + 1$  valeurs de blocs sont donc possibles. Pour une image de taille  $N_x \times N_y$ ,  $\frac{N_x \times N_y}{N_{max}^2}$  valeurs sont examinées : il en résulte  $\left(\log_2 \frac{N_{max}}{N_{min}} + 1\right)^{\frac{N_x \times N_y}{N_{max}^2}}$  combinaisons différentes de valeurs de blocs. Dans le cas de l'image "crosses" de taille  $256 \times 256$ , et d'une partition  $QP^{[16...2]}$ , il existe donc  $4^{256}$  combinaisons possibles de valeurs de blocs pour la seule première étape d'encodage, soient environ  $10^{154}$  combinaisons !

Comme la solution précédente ne peut être raisonnablement envisagée, nous avons testé l'algorithme de reconstruction de l'image en substituant à la grille originale, une grille uniforme composée uniquement de blocs de taille  $16 \times 16$ , afin de vérifier si l'image reconstruite reste pertinente. Les résultats obtenus sont visibles figure 8.8. Si un spécialiste peut reconnaître Lena par exemple, c'est essentiellement parce que le haut de la pyramide réalisée par les méthodes multirésolution du LAR (en l'occurrence l'Interleaved S+P) traite en une passe l'ensemble des pixels du niveau correspondant à l'image originale sous-échantillonnée par 16 dans les deux dimensions. Si nous décidons d'ajouter au moins un niveau de décomposition de la pyramide, l'information approximative résultante devient suffisamment peu cohérente pour qu'aucune image ne soit plus reconnaissable.

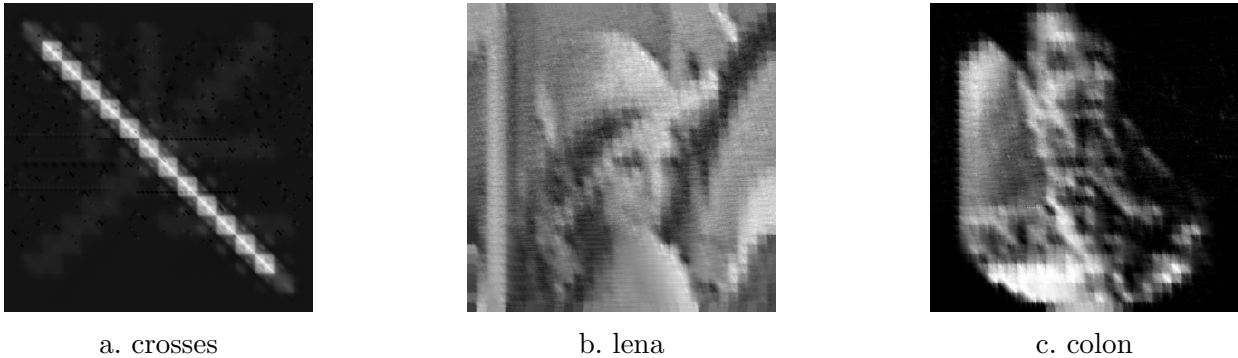


FIG. 8.8: Grille inconnue au décodeur : hypothèse d'une grille uniforme (blocs  $16 \times 16$ ).

## 3.2 Récupération incorrecte de la valeur d'une erreur de prédiction

### 3.2.1 Impact théorique d'une erreur

Nous cherchons dans un premier temps à évaluer de façon théorique la propagation de l'erreur dans les phases de prédiction. Considérons d'abord la toute première valeur transmise, à savoir celle encodant le premier pixel du niveau supérieur de la pyramide : du fait que le MICD mis en œuvre agisse selon un parcours classique en "raster" (ligne après ligne), l'erreur se propage sur l'ensemble de la décomposition. Ainsi, un décalage doit être observé sur l'image entière.

Pour les niveaux inférieurs, nous devons tenir compte du fait que la décomposition se fait selon deux pyramides entrelacées, tout en séparant l'information globale de l'information de texture. La figure 8.9 illustre la façon dont se propage une erreur réalisée sur la première ou la seconde diagonale d'un bloc  $2 \times 2$  au sein d'un même niveau de la pyramide, sans distinction de la nature des blocs.

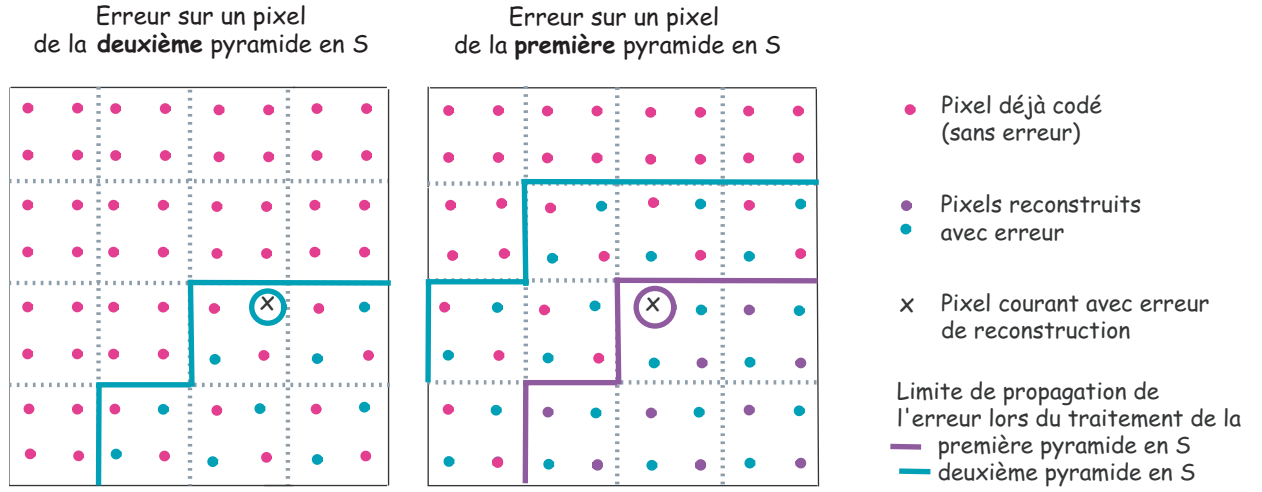


FIG. 8.9: Analyse théorique de la propagation intra-niveau des erreurs

Du fait de la prédiction réalisée à partir des blocs  $2 \times 2$  adjacents, dans le cas d'une erreur observée au niveau  $l$  de la pyramide au pixel  $Y_l(i, j)$  issu de la seconde pyramide en S, l'ensemble des pixels suivants seront affectés :

$$\begin{cases} Y_l(i + 2n_l, j + 2m_l), \\ Y_l(i - 1 + 2n_l, j + 1 + 2m_l), \end{cases} \quad (8.1)$$

avec

$$\begin{cases} 1 \leq n_l \leq \frac{N_x^l}{2} - i, & \text{et} & 1 \leq m_l \leq \frac{N_y^l}{2} - j; \\ \text{et} \\ -\frac{i}{2} \leq n_l \leq -1, & \text{et} & 1 \leq m_l \leq |n_l|. \end{cases} \quad (8.2)$$

Pour un pixel de la première diagonale en S, la succession des deux étapes de prédiction (première et deuxième diagonale d'un bloc, ou encore première et deuxième pyramide en S) élargit encore l'espace de propagation de l'erreur. Ainsi, dans un premier temps, la phase d'estimation de la première diagonale implique une propagation de l'erreur aux pixels

$$\begin{cases} Y_l(i + 2n_l, j + 2m_l), \\ Y_l(i + 1 + 2n_l, j + 1 + 2m_l), \end{cases} \quad (8.3)$$

avec les conditions données par les relations 8.2. Les pixels affectés par le traitement de la seconde diagonale sont quant à eux localisés aux positions suivantes :

$$Y_l(i - 1 + 2n_l, j - 2 + 2m_l) \quad \text{où} \quad \begin{cases} 1 \leq n_l \leq \frac{N_x^l}{2}, \\ 1 \leq m_l \leq \frac{N_y^l}{2}. \end{cases} \quad (8.4)$$

Si le niveau courant n'est pas le niveau 0 de la pyramide, l'erreur se propage à travers la structure pyramidale. Les pixels issus de la décomposition d'un pixel erroné  $Y_l'(i, j)$  au niveau supérieur sont naturellement à leur tour reconstruits par une valeur fausse. Une autre partie des



pixels seront quant à eux affectés du fait de la propagation intra-niveau décrite ci-dessus. Ainsi, chacun des pixels erronés  $Y'_{l-\alpha}(2^\alpha i, 2^\alpha j)$  d'un niveau  $l - \alpha$  donné, participe à cette propagation de telle sorte que les pixels

$$Y_{l-\alpha}(2^\alpha(i-1) + n_{l-\alpha}, 2^\alpha(j-1) + m_{l-\alpha}), \quad (8.5)$$

avec

$$0 \leq m_l \leq N_y^{l-\alpha} - i, \quad \text{et} \quad -2 \times \left\lfloor \frac{m_{l-\alpha}}{2} \right\rfloor \leq n_l \leq N_x^{l-\alpha} - i; \quad (8.6)$$

soient alors reconstruits par une valeur erronée.

### 3.2.2 Observation sur des cas réels

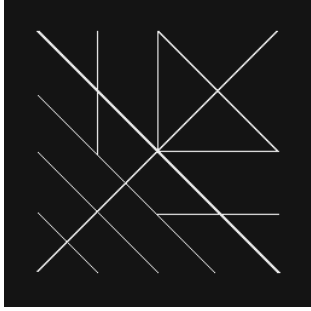
Les expérimentations ont été menées sur quatre images : "crosses", "lena", "colon" et "ct\_abdomen\_fr". Nous distinguons ici les cas d'une erreur de transmission produite sur une valeur du niveau supérieur de la décomposition pyramidale, d'une erreur sur les niveaux inférieurs, où sont traitées de façon séparée information globale (image LAR) et texture. Les pixels représentés en bleu dans les images d'erreurs signifient que la valeur reconstruite est égale à la valeur de l'imagette originale.

**Première valeur transmise : encodage du pixel  $Y_{l_{max}}(0,0)$ .** La figure 8.10 intègre les résultats des tests réalisés sur les quatre images : les valeurs erronées sont volontairement très éloignées de la valeur d'origine afin de visualiser au mieux leur effet. Outre l'image reconstruite, nous avons représenté les images de différence entre image d'origine et image décodée, afin de percevoir la propagation de l'erreur.

Pour "lena" et "ct\_abdomen\_fr", l'hypothèse de décalage opéré sur l'ensemble de l'image se vérifie. Le comportement est différent dans les cas de "crosses" et "colon" : le décalage n'est pas uniforme. L'explication réside dans le fait que ces deux images présentent un fort contraste : les valeurs reconstruites sont naturellement bornées par la dynamique de l'image (ici 8 bits par points), et la progression de l'erreur se voit ralentie dès lors qu'une borne est atteinte (voir 8.11).

Dans le cas d'images du type "colon" (typiquement IRM ou mammographies, cf annexe I), la première valeur à encoder de façon prédictive est égale à zéro. Or le haut de la pyramide est traité par un MICD simple, où la première valeur est estimée systématiquement par  $\check{Y}_{l_{max}}(0,0) = 128$ . Ainsi, l'erreur commise (et donc la valeur transmise) est égale à  $e_{l_{max}}(0,0) = -128$ . Si au décodeur la valeur récupérée  $\tilde{e}_{l_{max}}(0,0)$  est inférieure à la valeur originale  $e_{l_{max}}(0,0) = -128$ , par le jeu des troncatures, nous reconstruisons tout de même le pixel à 0, et par suite, aucune différence n'est observée entre l'image au décodeur et l'image originale.

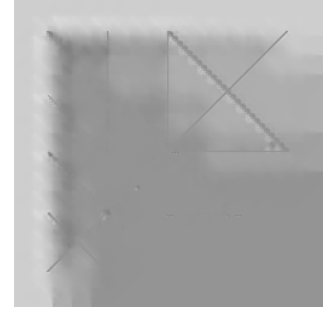
**Pixel quelconque au niveau  $l_{max}$  de la pyramide.** Les constatations générales précédentes s'appliquent dans le cas d'un pixel pris au hasard au niveau  $l_{max}$  de notre pyramide. Le schéma prédictif au niveau supérieur de la pyramide étant fondé sur l'observation restreinte du voisinage causal, l'erreur se propage majoritairement dans le quadrant de l'image en bas à droite dont le point d'origine est le pixel modifié. Les prédictions inter- et intra-niveaux successivement appliquées lors de la descente de la pyramide contribuent à élargir la zone erronée (voir figure 8.12).



A1. crosses : Image originale



A2. crosses : Image reconstruite



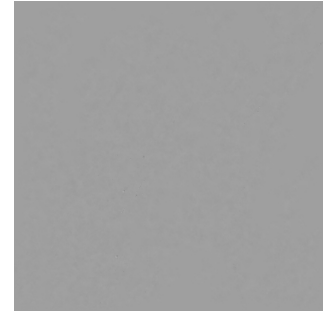
A3. crosses : propagation de l'erreur



B1. lena : Image originale



B2. lena : Image reconstruite



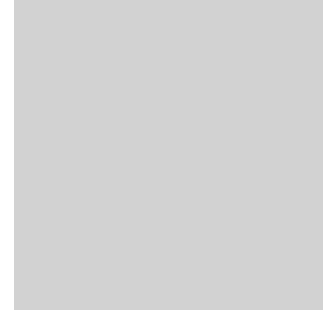
B3. lena : propagation de l'erreur



C1. ct.abdomen.fr : Image originale



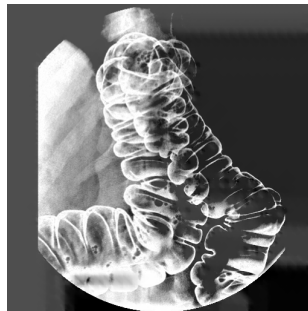
C2. ct.abdomen.fr : Image reconstruite



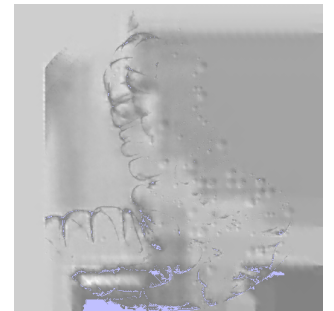
C3. ct.abdomen.fr : propagation de l'erreur



D1. colon : Image originale

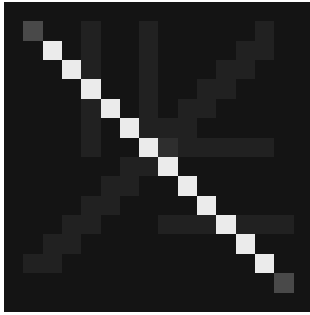


D2. colon : Image reconstruite

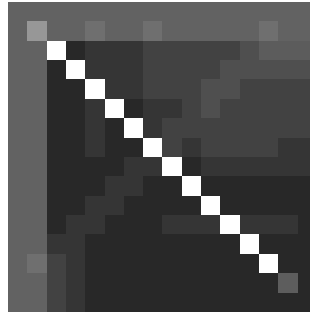


D3. colon : propagation de l'erreur

FIG. 8.10: Valeur d'erreur de prédiction erronée observée sur le pixel  $Y_{l_{max}}(0,0) = Y_4(0,0)$ .



a. Image niveau 4 au codeur



b. Image niveau 4 au décodeur



c. Propagation de l'erreur au niveau 4

FIG. 8.11: Propagation de l'erreur intra-niveau (image crosses, niveau  $l = l_{max} = 4$ ). Erreur sur la première valeur transmise.

**Pixel quelconque d'un niveau inférieur.** Du fait de la propagation certaine d'une erreur intra- et inter-niveaux, et de la structure pyramidale, il est évident que la perception visuelle de cette erreur est plus grande si elle est effectuée sur les niveaux les plus hauts de la pyramide. C'est pourquoi nous ne présentons essentiellement ici que les expérimentations réalisées sur le niveau  $l = l_{max} - 1 = 3$  de la décomposition.

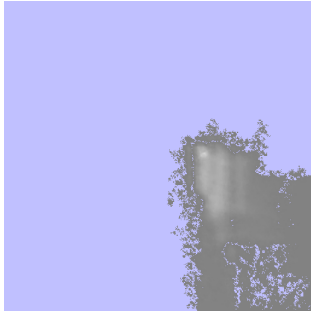
La figure 8.13 recense les trois cas de propagation de l'erreur lors de la première décomposition pyramidale (obtention de l'image LAR basse résolution). Les pixels affectés correspondent bien à ceux prévus dans notre étude théorique. Cependant, malgré la forte perturbation introduite, les dégradations restent très localisées, de telle sorte que les images résultantes à pleine résolution gardent globalement une qualité visuelle étonnante (figures 8.14 et 8.15). En fait, deux phénomènes entrent conjointement en jeu. D'une part, notre système prédictif opère une élimination naturelle des valeurs aberrantes, s'apparentant alors à un filtrage efficace (par exemple : filtrage médian pour la texture codée par la première pyramide en S). D'autre part, l'estimation d'une valeur de pixel tient compte d'un grand nombre de voisins, favorisant ainsi la "dilution" de la donnée erronée. Pour un même niveau de la décomposition, l'erreur se propagera d'autant plus qu'elle sera effective au niveau la première descente pyramidale.

La figure 8.16 permet de vérifier les faits suivants : d'une part, l'impact visuel est négligeable en cas de "faible" erreur commise sur une valeur reconstruite (image 8.16.A1,  $\hat{z}_1^{3,2} - z_1^{3,2} = 50$ ), et d'autre part, si la valeur erronée est obtenue dans les niveaux inférieurs de la pyramide, les déformations résultantes sont moindres. Enfin, l'erreur diffuse différemment selon qu'elle s'applique lors de la première ou de la deuxième descente de la pyramide : dans ce dernier cas, sa propagation reste relativement restreinte.

**L'ensemble des observations réalisées dans cette section confère à l'Interleaved S+P une propriété intéressante de robustesse vis-à-vis d'une erreur de transmission.**



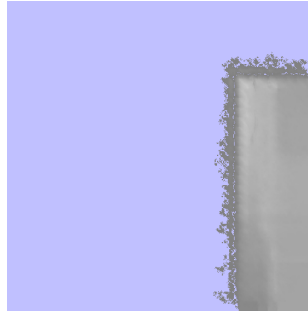
A1. lena : Image reconstruite



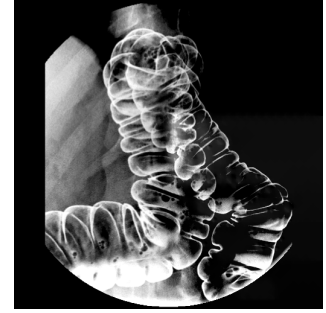
A2. lena : propagation de l'erreur



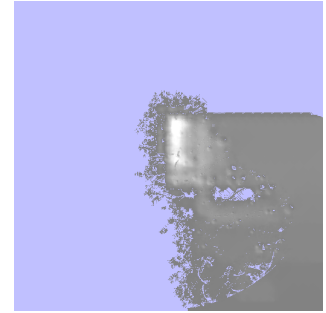
B1. abdomen : Image reconstruite



B2. abdomen : propagation de l'erreur



C1. colon : Image reconstruite



C2. colon : propagation de l'erreur

FIG. 8.12: Valeur d'erreur de prédiction erronée observée sur un pixel du niveau  $l_{max} = 4$ .



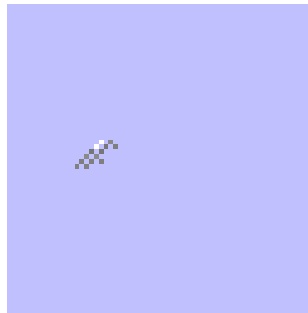
A1. Erreur sur  $z_1^{3,2}$ , gradient de la 1<sup>ère</sup> pyramide en S du niveau 3



A2. Propagation de l'erreur issue de A1.



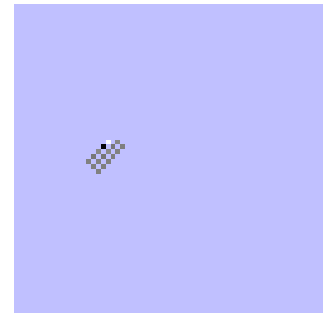
B1. Erreur sur  $z_0^{3,3}$ , moyenne de la 2<sup>nde</sup> pyramide en S du niveau 3



B2. Propagation de l'erreur issue de B1.



C1. Erreur sur  $z_1^{3,3}$ , gradient de la 2<sup>nde</sup> pyramide en S du niveau 3



C2. Propagation de l'erreur issue de C1.

FIG. 8.13: Observation de la propagation intra-niveau de l'erreur.

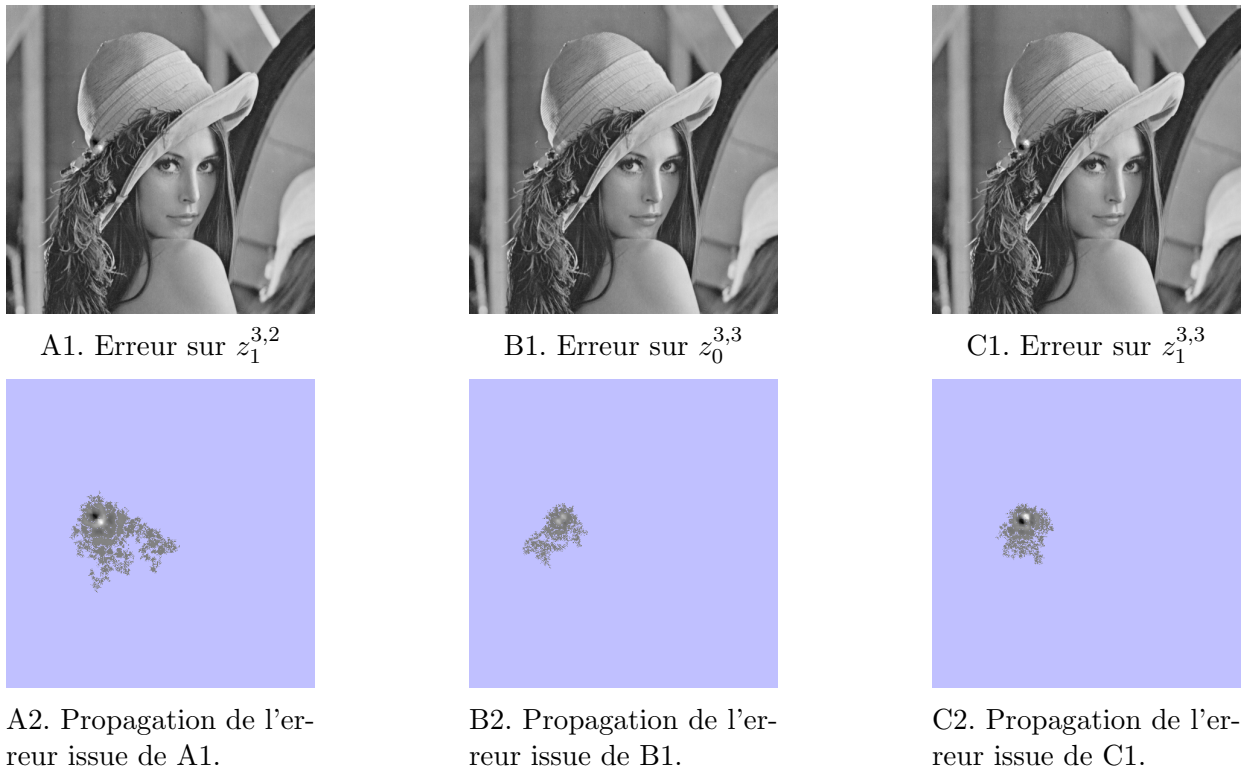


FIG. 8.14: Observation de la propagation complète de l'erreur (intra- et inter-niveau) appliquée sur l'image LAR basse résolution : décomposition totale de la pyramide.

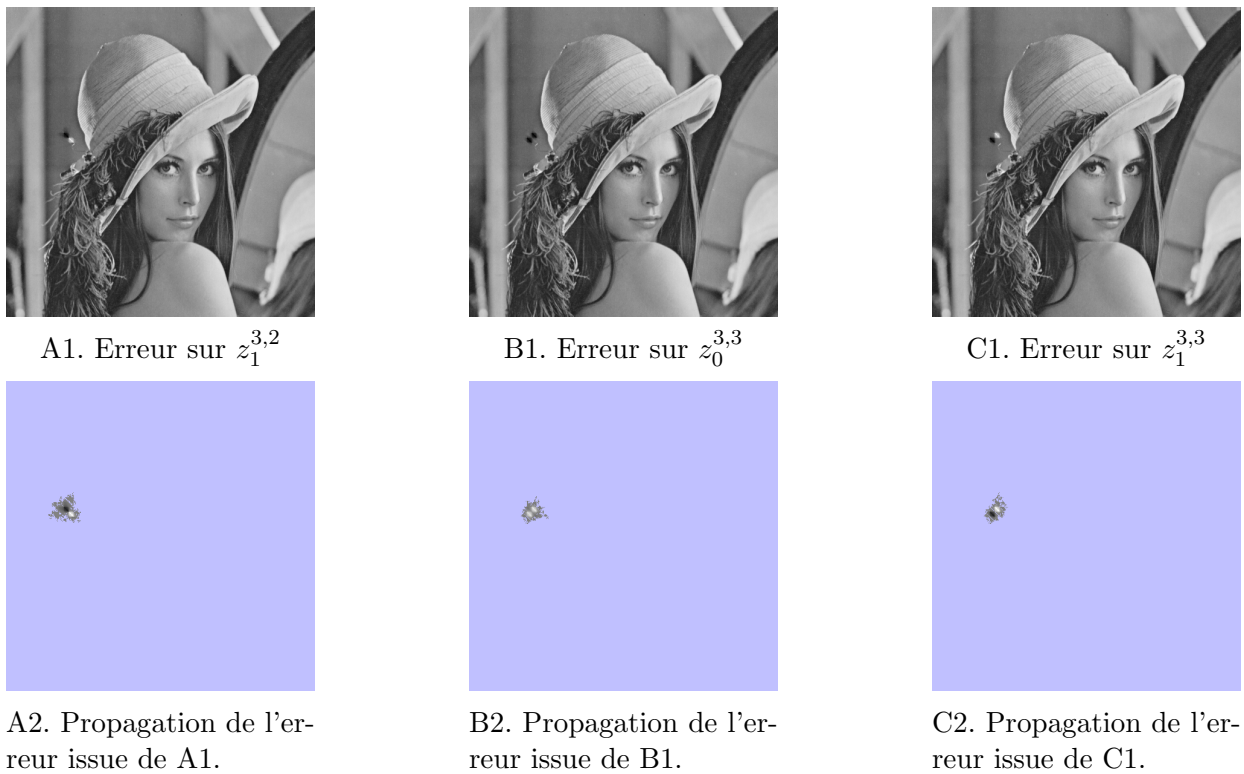


FIG. 8.15: Observation de la propagation complète de l'erreur (intra- et inter-niveau) appliquée sur l'information de texture : décomposition totale de la pyramide.



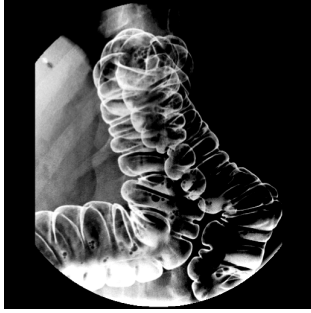
a. Faible erreur sur  $z_1^{3,3}$  - image LAR



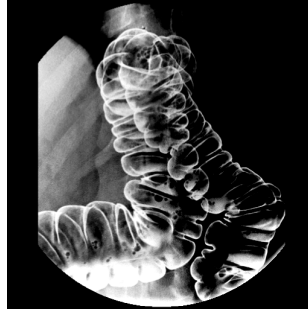
b. Erreur sur  $z_1^{1,2}$  - image LAR



c. Erreur sur  $z_1^{1,2}$  - texture



d. Erreur sur  $z_1^{3,2}$  - image LAR.



e. Erreur sur  $z_1^{1,2}$  - image LAR.



f. Erreur sur  $z_1^{1,2}$  - texture.



d. Erreur sur  $z_1^{3,2}$  - image LAR.



e. Erreur sur  $z_1^{1,2}$  - image LAR.



f. Erreur sur  $z_1^{1,2}$  - texture.

FIG. 8.16: Exemple d'images reconstruites après réception d'une valeur erronée.

### 3.3 Remarques

Suite aux expérimentations présentées ci-dessus, relevons le fait suivant : notre implantation se révèle résistante aux erreurs de transmission. Ainsi, l'algorithme donne une image résultat, quelle que soit la situation rencontrée (erreur sur la grille, sur une erreur de prédiction à tout niveau de la pyramide). La robustesse évoquée au paragraphe précédent ne se limite pas aux seules performances visuelles de l'algorithme : le qualificatif s'applique aussi à la conception même de l'Interleaved S+P.

Nous avons également évoqué les possibilités de cryptographie liée à une utilisation particulière de l'information de grille (§3.1). Cette propriété liée à la sécurisation de l'image peut s'étoffer par des procédés de **tatouage**, où les données cachées s'insèrent dans les zones les moins visibles et sans répercussion visuelle dommageable pour le reste de l'image.

Enfin, une thèse et un stage Master Recherche, débutant à cette date en collaboration avec le laboratoire SIC de Poitiers, devraient permettre de répondre rapidement aux préoccupations de robustesse de la chaîne de codage global. Nous envisageons pour ce faire de réorganiser les flux transmis et d'ajouter des éléments de synchronisation (sur l'exemple de MPEG-4), ou encore d'inclure un codeur entropique robuste à code symétrique adapté à notre algorithme.

## 4 La transformée Mojette-Dirac

Les travaux réalisés sur la transformée Mojette ont donné lieu à de nombreuses publications, dont [Par01, Gué01, GN03, PNG03]. Dans cette section, nous nous focaliserons sur l'aspect technique de description multiple de la méthode. Il existe de nombreuses variantes de la transformée Mojette : transformée Mojette-Haar, Mojette-Spline... Celle présentée dans ce chapitre est dénommée transformée Mojette-Dirac : nous verrons brièvement la définition de la transformée directe, ses propriétés remarquables, ainsi que la méthode de reconstruction de l'image à partir des données Mojette.

### 4.1 Transformée directe

La transformée Mojette est une transformée de Radon exacte et discrète. Elle permet de décrire aisément une image (support 2D) par un ensemble fini de projections unidimensionnelles (vecteurs 1D). Chaque angle de projection  $\theta$  est défini par un couple d'entiers  $(p, q)$ , premiers entre eux, où  $\frac{q}{p} = \tan \theta$ . La transformée Mojette directe d'une image  $f(k, l)$ , notée  $\mathcal{M}f$ , est représentée par l'ensemble de  $N$  projections  $\mathcal{M}_{p,q}f$ , de telle sorte que

$$\mathcal{M}f = \{proj_{p_i, q_i}, i = 1, 2, \dots, N\}. \quad (8.7)$$

Le calcul de cette transformée ne nécessite que des additions et des soustractions. Ainsi, chaque projection est constituée d'un ensemble d'éléments appelés *bins*, de valeur égale à la somme des pixels  $f(k, l)$  situés sur la droite  $m = -qk + pl$ .

La transformée Mojette-Dirac s'écrit

$$\begin{cases} [\mathcal{M}_{p,q}f](m) = proj(p, q, m) = proj_{p,q}(m) = proj_{\theta}(m), \\ [\mathcal{M}_{p,q}f](m) = \sum_k \sum_l f(k, l) \Delta(m + qk - pl), \end{cases} \quad (8.8)$$

où  $\Delta$  représente la fonction de Kronecker<sup>9</sup>. Les figures 8.17 et 8.18 proposent une illustration de la construction de deux projections Mojette d'un support de taille  $4 \times 4$ .

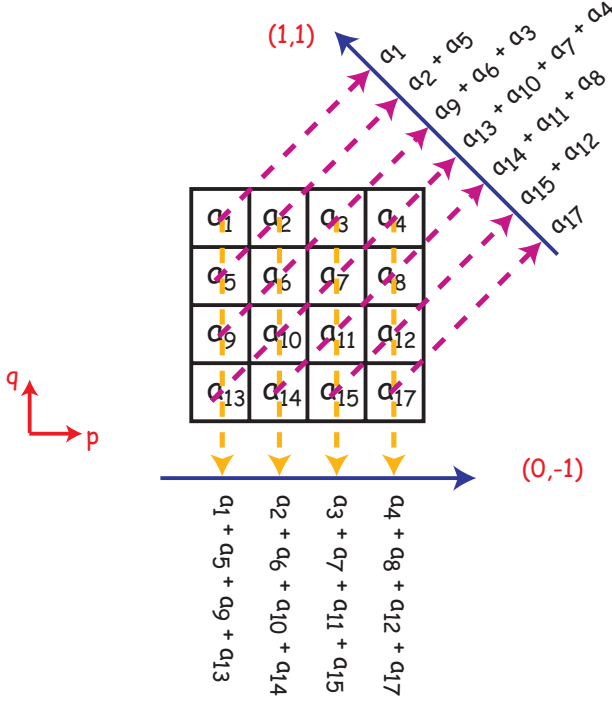


FIG. 8.17: Transformée Mojette directe : construction des projections.

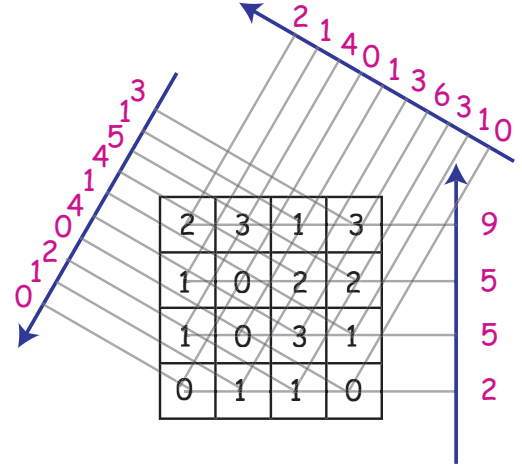


FIG. 8.18: Exemple d'application de la transformée Mojette selon les directions  $(1, 0)$ ,  $(1, 2)$  et  $(-2, 1)$

## 4.2 Propriétés de la transformée.

Pour une projection donnée, tout pixel contribue pour un et un seul *bin*. De ce fait, si l'image est de taille  $P \times Q$ , la complexité d'une projection est de l'ordre de  $\mathcal{O}(PQ)$ . Ainsi, la complexité globale de l'algorithme est en  $\mathcal{O}(PQN)$ , où  $N$  est le nombre de projections.

La transformée Mojette est donc linéaire selon la taille de l'image (nombre de pixels), et le nombre total de projections effectuées. Cette propriété s'apparente au comportement des transformées de Fourier et de Meyer (ondelettes). Toutefois, à l'opposé des ondelettes, les transformées de Radon et Mojette n'ont pas pour but de réaliser une décomposition orthogonale du signal, mais plutôt une *frame*<sup>10</sup>. En conséquence, ce type de transformée produit une redondance de symboles utile à la reconstruction. Cette propriété constitue en fait la différence fondamentale observable entre les deux familles de transformées. De plus, si les transformées de Fourier et de Meyer ordonnent naturellement les coefficients de façon hiérarchique, les *bins* issus de l'application de la Mojette ont tous le même poids dans le domaine transformé. Une projection donnée apporte

$${}^9\Delta(m) = \begin{cases} 1 & \text{si } m = 0, \\ 0 & \text{autrement.} \end{cases}$$

<sup>10</sup>Frame : soit  $\mathcal{F}$  un opérateur linéaire de  $\mathbb{R}^N$  vers  $\mathbb{R}^M$  défini par  $(\mathcal{F}x)_n = \langle x, \varphi_n \rangle$ , pour  $n = 1, \dots, M$ . Un ensemble  $\{\varphi_n\}_{n=1}^M \subset \mathbb{R}^N$  est une *frame* de  $\mathbb{R}^N$  si il existe deux constantes  $A > 0$  et  $B < \infty$  telles que  $A\|x\|^2 \leq \sum_{n=1}^M |\langle x, \varphi_n \rangle|^2 \leq B\|x\|^2$ .  $\mathcal{F}$  est alors un opérateur de *frame*.



alors une quantité d'information utile proportionnelle à sa capacité de reconstruire une partie du domaine original. Enfin, chaque projection contient la valeur moyenne du signal (somme de tous les *bins*) ; dans le cas des transformées orthogonales, cette information est toujours uniquement présente dans le premier coefficient.

L'implantation de la transformée Mojette directe et inverse tire parti de cette propriété de redondance inhérente aux *frames*, à des fins de description et de transmission d'images : l'ordre de construction et de reconstruction est alors laissé au choix de l'utilisateur.

### 4.3 Transformée inverse

#### 4.3.1 Algorithme de reconstruction

La transformée Mojette peut s'écrire comme le produit de la matrice transformée  $M$  avec le vecteur  $F$  composé des valeurs de pixels. Inverser la transformée Mojette revient alors à résoudre le système linéaire  $M \cdot F = B$ , où  $B$  contient les valeurs des *bins*. Cependant, la matrice de transformation  $M$  est de taille conséquente (nombre de pixels  $\times$  nombre de *bins*), rectangulaire et, le résultat de l'application de la Mojette s'apparentant à une *frame*, très creuse. C'est pourquoi le calcul efficace de la transformée inverse ne s'appuie pas sur le formalisme matriciel.

L'algorithme de reconstruction de la Mojette découle de l'observation suivante : tous les *bins* ne font pas intervenir le même nombre de pixels projetés. Il est évident que la valeur d'un *bin* associé à un unique pixel est égale à la valeur de ce pixel. Lorsqu'une telle correspondance univoque est trouvée, la valeur du pixel prend celle du *bin*. Ce pixel est alors retiré de l'ensemble des projections : sa valeur est soustraite à tous les *bins* sur lesquels il a été projeté. A ce stade, les *bins* restant représentent la transformée Mojette de la partie non reconstruite de l'image.

Ainsi, la reconstruction globale de l'image à partir d'un ensemble de projections s'effectue par l'itération des étapes

1. recherche d'une correspondance univoque (*bin*, pixel), *i.e.* un *bin* issu de la projection d'un unique pixel),
2. "retro-projection" de sa valeur sur le pixel original (affectation directe de la valeur du *bin* à la valeur du pixel considéré),
3. réévaluation des projections,

jusqu'à reconstruction complète de l'image, ou jusqu'à l'absence de correspondance univoque entre *bin* et pixel. Dans ce dernier cas, seule une partie du support est reconstruite.

Notons que la transformée inverse est de complexité équivalente à celle de la transformée directe : elle s'avère elle aussi linéaire en taille de l'image et en nombre de projections.

### 4.4 Critère de reconstruction et redondance - Propriétés de reconstruction

Les conditions de reconstruction d'un support rectangulaire ont été données par Katz [Kat78].

**Lemme 8.2** (*Lemme de Katz*) Une image rectangulaire de taille  $P \times Q$  est reconstructible à partir d'un ensemble  $K$  de projections  $\{(p_i, q_i)\}$  si et seulement si

$$\sum_{i=1}^K |p_i| \geq P \quad \text{ou} \quad \sum_{i=1}^K |q_i| \geq Q. \quad (8.9)$$

Ainsi, l'utilisateur peut choisir d'observer indifféremment l'une ou l'autre condition afin de construire un ensemble réversible de projections. Par exemple, posons pour chaque projection  $i$ ,  $q_i = 1$ . Alors une image  $P \times Q$  est reconstituée si  $Q$  projections sont réalisées, et ce quelle que soit la valeur de  $P$  !

De cet ensemble réversible de  $Q$  projections, il est possible d'obtenir un ensemble **redondant** de  $N$  projections ( $N > Q$ ) par la construction de nouvelles projections. Grâce à ces  $N - Q$  projections supplémentaires, **tout sous-ensemble de  $Q$  projections parmi les  $N$  suffit à la reconstruction de l'image complète** (autrement dit, la matrice de transformation est de rang  $Q$ ). De ce fait, si l'une des projections (ou simplement une partie cette projection) est manquante ou encore contient beaucoup d'erreurs binaires, il est possible de la remplacer par toute autre projection redondante.

De plus, l'ordre de réception des données au décodeur n'influe aucunement sur la reconstruction du signal : le processus de décodage s'enclenche dès que le nombre de données reçues est suffisant.

## 4.5 Description multiple

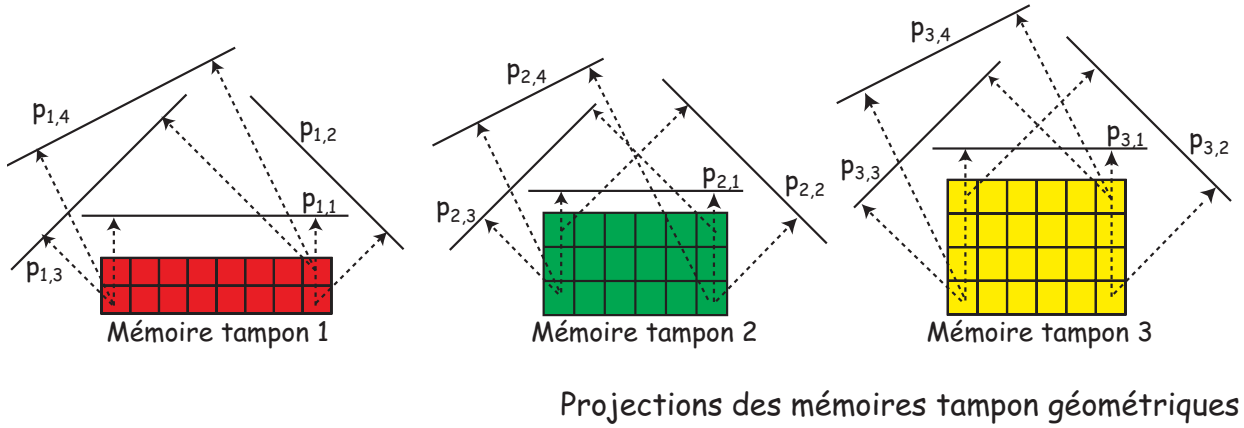
La mise en œuvre d'un système de communication, à garantie de qualité de service en termes de pertes de paquets, peut être réalisée par des codes à description multiple [Goy01]. Le principe général commun à l'ensemble de ces codes consiste à séparer la source originale en un ensemble indépendant de descriptions, optimisant à la fois les possibilités de reconstructions partielles (versions dégradées de l'image originale) et la redondance introduite. En fait, le codage à description multiple (MD) est contraint au compromis suivant : produire des descriptions toutes individuellement de bonne qualité mais suffisamment différentes entre elles.

Dans l'optique de pouvoir reconstruire les  $N$  paquets originaux à partir de tout sous-ensemble de  $N$  paquets parmi  $M$ , les codes MDS (Maximum Distance Separable), et en particulier les codes Reed-Salomon font partie des techniques classiques. La transformée Mojette peut être facilement employée dans ce contexte [PNG03]. Dans cette section, nous introduisons le mécanisme pilotant la définition des paquets, et convertissant dans le même temps le flux LAR multirésolution à priorité en un flux MD non prioritaire de  $N$  paquets.

Chaque élément du flux de données compressées par le LAR (chaque sous-flux défini au préalable) peut être représenté par un support 2D. Ces supports s'apparentent à une mémoire tampon géométrique (ou buffer géométrique) du système de communication. L'application de la transformée Mojette directe sur chacun de ces buffers produit une description multiple de l'information. Chaque description dans le flux MD occupe un paquet réseau entier ; les termes paquets et projections sont donc interchangeables.

La méthode utilise plusieurs buffers géométriques qui peuvent être concaténés afin d'optimiser le "facteur d'étirement" (*stretch factor*<sup>11</sup>) du code. Le partitionnement des données est ici extrêmement simple : il fait intervenir une mémoire tampon pour chaque sous-flux associé à une priorité. De ce fait, la transformée Mojette est appliquée et calculée sur tous les supports de

<sup>11</sup>Stretch factor : soit  $X = \{x_1, x_2, \dots, x_n\}$  un flux encodé par un code MDS en un ensemble de symboles  $Y = \{y_1, y_2, \dots, y_m\}$ . Le facteur d'étirement (*stretch factor*) est donné par  $\lambda = \frac{m}{n}$ .



Paquet 1 = en-tête +  $[p_{1,1}, p_{2,1}, p_{3,1}]$



Paquet 2 = en-tête +  $[p_{1,2}, p_{2,2}, p_{3,2}]$



Paquet 3 = en-tête +  $[p_{1,3}, p_{2,3}, p_{3,3}]$



Paquet 4 = en-tête +  $[p_{1,4}, p_{2,4}, p_{3,4}]$



### Mise en paquet

FIG. 8.19: Schéma à trois niveaux de protections avec séparation en mémoires tampon géométriques. La construction d'un paquet est réalisée par la concaténation de projections du même angle.  $p_{a,b}$  représente la  $b^{me}$  projection du  $a^{me}$  buffer géométrique.

façon simultanée. Cette configuration particulière conduit à la mise en œuvre des mêmes angles de projections pour chaque buffer traité, préalable à la construction d'une trame réseau.

La figure 8.19 illustre la méthode. Trois supports rectangulaires sont définis avec des propriétés de reconstruction distinctes afin d'obtenir trois niveaux scalables de protection. Les sous-flux 1, 2 et 3 sont répartis respectivement dans les mémoires tampon géométriques 1, 2 et 3. Le support 1 (resp. 2 et 3) est reconstituible au moyen de deux projections (resp. trois et quatre) issues de l'ensemble des projections  $S = \{(0, 1), (1, 1), (-1, 1), (-2, 1)\}$ . Le paquet 1 (resp. 2, 3 et 4) contient alors toutes les projections d'angle (0, 1) (resp. (1, 1), (-1, 1) et (-2, 1)). Ces considérations impliquent une taille variable de paquets tout en simplifiant l'insertion des étiquettes affectés aux angles.

Cette méthode donne une solution protégeant la scalabilité de la source. La transformée Mojette produit des descriptions équivalentes aux données nécessaires à la transmission sur réseaux. Pour un seul niveau de priorité, si la transmission est composée de  $M$  projections plus  $N - M$  projections redondantes, il est possible de décoder de façon déterministe un niveau de reconstruction à partir de n'importe quels  $M$  paquets reçus parmi les  $N$  transmis. Grâce à ce

mécanisme, nous assurons la hiérarchie intrinsèque de la source à travers un canal "best-effort"<sup>12</sup>, avec une garantie de Qualité de Service menée de bout-en-bout.

## 5 Le LAR aux Mojettes : codage conjoint source-canal

Les applications de télémédecine nécessitent à la fois un algorithme de compression efficace, et une transmission sécurisée des données. L'introduction d'une méthode progressive de codage améliore sensiblement les performances du procédé de sécurisation : les octets supplémentaires alloués à l'information redondante sont plus nombreux dans le cas des schémas non progressifs. Dans ce contexte, nous décrivons dans cette section la technique de codage d'image conjointe LAR et Mojette. Si l'Interleaved S+P révèle des performances extrêmement intéressantes dans le domaine de l'imagerie médicale (voir chapitre 6), nous avons dans un premier temps, pour cette étude, mis en place le LAR-APP (pour des raisons historiques, le LAR-APP étant opérationnel avant l'Interleaved S+P!).

### 5.1 Constitution des paquets IP

A l'encodage d'un ensemble de  $i$  flux associés à des niveaux de priorité distincts, le nombre maximal de projection, à savoir  $N$  (nombre de paquets IP), doit être au préalable déterminé. La transformée Mojette est alors exécutée sur chacun des sous-flux  $i$ , connaissant le nombre de projections nécessaires  $H_i$ . Par conséquent, il existe  $N - H_i$  projections redondantes pour chaque flux  $i$ . L'étape finale consiste en la concaténation de toutes les projections.

Sous-flux	1	2	3	4	5	6
Taille initiale des flux (octets)	1182	907	2168	3824	13010	44943
Largeur des supports (octets)	394	227	434	638	1859	5618
Nombre de projections nécessaires ( $H_i$ )	3	4	5	6	7	8
Nombre de projections redondantes ( $N - H_i$ )	5	4	3	2	1	0

TAB. 8.1: Taille des différents niveaux de reconstruction (*layer*) et le nombre de projections nécessaires et redondantes associées. Image "angioma" traitée par le LAR-APP (3.35 bpp)

Le flux global compressé par le LAR est tout d'abord séparé en six sous-flux successifs, dont les tailles sont répertoriées dans le tableau 8.1. Le premier flux comporte à la fois l'information codant pour la grille et pour le niveau supérieur de la pyramide (niveau 4). Les flux 2,3 et 4 correspondent respectivement aux niveaux 3,2 et 1 de la première descente de pyramide (récupération de l'image des blocs). L'ajout du sous-flux 5 permet la reconstruction de l'image à résolution moitié (texture plaquée sur la précédente image des blocs). Enfin, le sixième sous-flux encode sans perte l'information restante à pleine résolution.

Les informations collectées dans le tableau 8.1, à savoir la taille de chaque support et le nombre de projections associées, sont représentées sur la figure 8.20. Nous repérons aisément le fait que la priorité la plus haute a été attribuée au premier sous-flux. En effet, comme il est évident que la grille et le niveau le plus haut de la pyramide sont les deux informations indispensables à toute

<sup>12</sup>Best-effort - effort maximum : Classe de qualité de service (QoS class) dans laquelle aucun paramètre spécifique de trafic, ni aucune garantie absolue de délivrance ne sont assurés.

reconstruction de l'image au décodeur (voir section 3), la protection maximale est naturellement allouée à ce sous-flux. Le processus de sécurisation se propose ainsi de sur-représenter ces données sensibles. Le niveau de protection diminue au fur et à mesure que nous avançons dans les étapes de décodage.

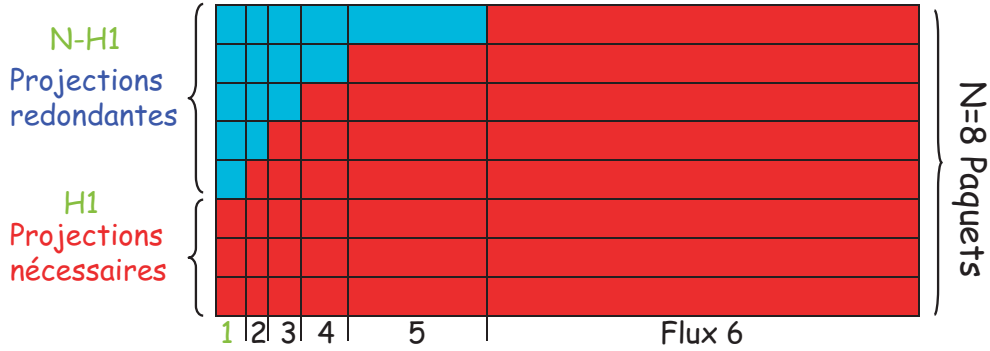


FIG. 8.20: Représentation des paquets IP : visualisation de la taille des paquets, nombre de projections nécessaires (en rouge) et nombre de projections redondantes (en bleu) par sous-flux. Image "angioma" traitée par le LAR-APP.

Dans l'exemple traitant de l'image Angioma par le LAR-APP, nous avons défini  $N = 8$  le nombre de projections par support, ou encore le nombre de paquets à transmettre. Les 1182 octets du premier flux sont représentés par un rectangle  $3 \times 394$ . Ce système d'encodage à priorité permet la reconstruction du niveau 4 de la décomposition multirésolution dès lors que  $H_1 = 3$  paquets IP parmi les 8 envoyés sont reçus au décodeur.

## 5.2 Protection inégale ou différenciée - Qualité image

L'observation des symboles de données, plutôt que celle du nombre de paquets, indique que la transformée Mojette est très proche de la reconstruction optimale : en effet, la Mojette ne nécessite que 2% de symboles en plus du volume de données initial pour un décodage intégral de l'image<sup>13</sup>. La redondance ici introduite apporte un surcoût égal à 7326 octets, *i.e.* 11% du volume global de données compressées, distribué sur les 8 paquets IP. La **redondance à priorité** permet ainsi de garantir la reconstruction de l'image issue du premier flux (niveau 4) si au moins 3 paquets sur 8 (37.5%) arrivent à destination, le niveau 3 pour 4 paquets sur 8, etc. Cette **stratégie de protection différenciée** apporte un véritable progrès. En effet, dans le cas d'une protection indifférenciée et pour un même surcoût (7326 octets), le volume initial (l'image codée sans perte) se représente alors sous la forme d'un rectangle de taille  $9 \times 7326$ . Il en résulte 10 projections, dont une unique projection redondante.

La figure 8.21 montre les niveaux de qualité de l'image reconstruite en fonction du taux de réception. Dans cette première version, seul le PSNR a été utilisé : nous ne l'utilisons pas ici comme critère de comparaison, mais comme moyen objectif de remarquer l'accroissement de la qualité

<sup>13</sup>Si le volume initial consiste en  $k$  symboles, l'inversion de la Mojette ne requiert que  $(1 + \varepsilon)k$  symboles, avec  $\varepsilon = 0.02$

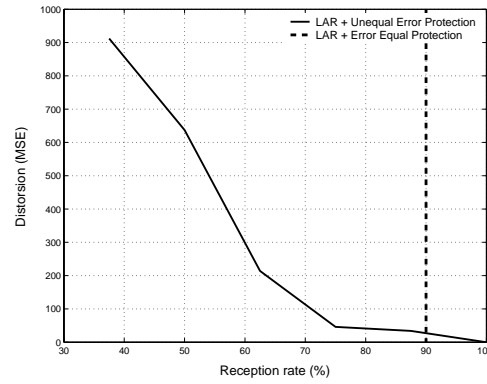


FIG. 8.21: Qualité de l'image reconstruite pour différent taux de réception. 100% signifie que toutes les projections (tous les paquets) ont été reçues.

en fonction de l'augmentation du nombre de paquets reçus. En collaboration avec Benoît Parrein et Patrick Le Callet (IRCCyN, Nantes), des travaux futurs doivent porter sur l'optimisation de l'algorithme de protection différenciée par la prise en compte automatique de la qualité de l'image reconstruite, qualité éprouvée au travers de métriques élaborées par P. Le Callet [CB03].

Pour une protection indifférenciée, la réception de 90% des données (9 paquets sur 10) assure la reconstruction intégrale de l'image originale. Cependant, la perte de plus d'un paquet lors de la transmission sur Internet empêche toute visualisation (même partielle) d'image au décodeur. Cet état de fait est évidemment inacceptable dans le contexte de la médecine d'urgence ! C'est pourquoi, la mise en œuvre d'une technique scalable conjointement avec un système de protection différenciée contre les erreurs (tel qu'il a été décrit dans ce chapitre), s'adapte parfaitement aux besoins de la télémédecine.

## 6 Apports de la méthode dans le cadre de la télémédecine

### 6.1 Réflexions autour du LAR

En premier lieu, les résultats affichés au paragraphe 5.3 du chapitre 6 montrent clairement l'efficacité de l'algorithme Interleaved S+P vis-à-vis des images médicales. Cette première constatation très simple justifie ainsi pleinement l'emploi de notre méthode. Par ailleurs, l'ensemble des fonctionnalités avancées décrites dans les chapitres précédents doivent être ici appréciées dans le cadre d'une utilisation médicale.

Pourquoi utiliser une approche multirésolution (le LAR-APP ou l'Interleaved S+P) dans un contexte de télémédecine ? La question mérite ici d'être posée. Même si les réseaux à haut-débits se développent et deviennent accessibles aux particuliers, il reste néanmoins très intéressant de transmettre une information au contenu organisé de façon hiérarchique, de telle sorte que l'utilisation de ces données par l'observateur final soit optimisée : par exemple, un pré-diagnostic peut être établi sur une image codée à bas débit, puis l'évaluation finale de la pathologie est réalisée à la réception complète de l'image. Cette hiérarchie intrinsèque à la méthode LAR peut être vue comme une métadonnée destinée à l'amélioration des échanges d'informations.

Cette constatation ne se cantonne pas à la notion de transmission de l'image, mais peut aussi s'étendre au cas de l'archivage. En effet, il est facile d'imaginer que de nombreuses requêtes

adressées à une base de données ne nécessitent comme réponse qu'une version très simplifiée de l'image (niveaux hiérarchiques les plus hauts). Dans le cadre du PACS en particulier, un médecin consultant un dossier patient, afin de lever (ou confirmer) tout soupçon de lésion, ne désire pas observer l'ensemble des images disponibles à haute résolution ! La sélection des images pertinentes par l'observation de leurs versions dégradées est beaucoup plus efficace.

Par ailleurs, grâce à la représentation en régions auto-extractibles (voir chapitre 2), il est très aisé de décrire une région d'intérêt, aussi bien au codeur qu'au décodeur. Ainsi, il est possible d'envisager (au moins !) deux cas d'utilisation optimisée. Un médecin sollicitant un confrère demande une confirmation de diagnostic : sur l'image à basse résolution (en qualité) produite par le LAR, il sélectionne la zone sensible pour transmettre l'information correspondante encodée localement sans perte (ajout de niveaux hiérarchiques dans le processus d'acheminement des données). Dans un autre contexte, un médecin accède à un examen antérieur : connaissant la région de l'image qui l'intéresse, il souhaite obtenir la plus haute résolution possible pour cette zone, le reste de l'image ne servant que de contexte à l'étude.

Ces "labels" associés à la hiérarchie, utilisés en tant que métadonnées, permettent l'accélération des requêtes et l'extraction de l'information. Les approches pyramidales du LAR ordonnent naturellement l'information compressée, et trouvent leur application pour de nombreuses tâches médicales.

## 6.2 Intérêt de la Mojette

Afin de justifier l'utilisation de la Mojette, en réponse aux contraintes du canal de transmission, il faut examiner l'intérêt d'une protection aux erreurs différenciée (UEP<sup>14</sup>) au vu d'une protection non différenciée (EEP<sup>15</sup>).

Pour l'acheminement d'un paquet IP, la mise en œuvre de codes correcteurs avant (codes FEC<sup>16</sup>) du type Reed-Salomon autorise la perte d'unités de transport sans toutefois compromettre l'interprétation de l'information reçue. Leur capacité de correction pour le flux entier peut, sous certaines conditions, suivre la propriété suivante : dès que  $M$  paquets parmi les  $N$  contenant les  $M$  messages utiles sont reçus au décodeur, il est possible de reconstruire le flux original de façon déterministe. Nous définissons l'EEP par un unique niveau de redondance pour l'ensemble du flux : le décodage échoue alors si plus de  $N - M$  paquets sont perdus. Lorsque la reconstruction exacte s'avère impossible (trop de pertes de données), il est cependant plus profitable d'essayer de compenser les erreurs plutôt que de les corriger. Pour ce faire, les schémas UEP sont capables de fournir une version dégradée de l'image, même lorsque la transmission est particulièrement perturbée. Par exemple, les codes FEC sont implantés suivant différents facteurs d'étirement à des fins de transmission de flux compressés par JPEG 2000 sur réseaux sans fil : l'UEP par ces codes FEC équivaut à une description multiple.

Dans le domaine de la protection différenciée, la contribution de la transformée Mojette, relativement aux FEC traditionnels du type Reed-Salomon, est l'ajustement des niveaux de priorité et donc de la redondance : la transformée offre ainsi une plus grande flexibilité. De ce

---

<sup>14</sup>UEP : Unequal Error Protection

<sup>15</sup>EEP : Equal Error Protection

<sup>16</sup>FEC : Forward Error Correcting



fait, la Mojette intègre au mieux la hiérarchie complexe proposée par le codage LAR.

## 7 Conclusion

Dans un premier temps, nous avons examiné dans ce chapitre les principes du PACS, dédié à l'archivage et la transmission des données médicales. Nous avons en particulier évoqué les préoccupations liées à la transmission sécurisée des images.

Dans ce cadre, nous avons procédé à l'analyse de la sensibilité au bruit du schéma de codage Interleaved S+P décrit au chapitre 6. En dépit de l'architecture complexe de l'ensemble de la méthode, nous avons pu souligner le **caractère robuste de l'Interleaved S+P vis-à-vis des erreurs de prédiction** : cette propriété se révèle extrêmement intéressante dans le cadre de la télémedecine. En effet, vue la faible propagation spatiale d'une erreur de transmission, la reconstruction de l'image apparaît suffisante pour une interprétation correcte, si toutefois la partie de l'image corrompue ne constitue pas une zone sensible : le tatouage et la stéganographie constituent deux exploitations naturelles de cet état de fait. Par ailleurs, **la sensibilité de la méthode envers le codage de la grille** présente elle aussi un avantage non négligeable dans le cadre d'une sécurisation des données. En effet, cette information de petite taille se révèle facile à crypter.

Un **codage conjoint LAR-Mojette** a par la suite été présenté à des fins de transmissions sécurisées d'images compressées sur les réseaux du type Internet. L'application comporte un codeur **scalable** LAR, dont les données compressées résultantes passent par une transformation Mojette. La Mojette s'apparente alors à un système d'**encodage à priorité**. L'association du LAR avec ce système fournit une **garantie de Qualité de Service** menée de **bout-en-bout**.

Du point de vue codage d'images, la technique du LAR-APP a été utilisée. De nouvelles mesures, prenant en compte l'Interleaved S+P (meilleure compression pour une même qualité progressive d'images), sont en cours d'évaluation.

La transformée Mojette est appliquée sur chaque sous-flux compressé issu du codeur LAR : une protection hiérarchique par contrôle du coût est ainsi définie. Toutes les projections et la redondance supplémentaire associée sont réparties entre tous les paquets IP. De ce fait, comme le coût de la Mojette ne dépend que du nombre de paquets reçus, et non par de leur contenu, le décodeur lance immédiatement le processus de reconstruction à la réception d'un paquet.

Ce travail constitue un prélude à un ensemble de projets communs à notre laboratoire et à celui de l'IRCCyN. Outre l'amélioration de la méthode par la mise en œuvre de techniques avancées (évaluation intelligente de la qualité des images, nouvelle approche Mojette, utilisation de l'Interleaved S+P ou RWHT+P), de nombreuses extensions à ce codage conjoint LAR-Mojette sont envisagées. En particulier, une thèse en co-tutelle a débuté cette année à l'IETR (Jean Motsch-Pivette) et porte sur les aspects liés à la sécurisation des données médicales (images 2D et volumes d'images encodés par le LAR), via la cryptographie, la stéganographie et le tatouage réalisés par la transformée Mojette.

En outre, il est à noter que nous avons réalisé une plate-forme de démonstration de la compression d'images par le LAR, destinée en particulier à un public concerné par la télémedecine. Cette plate-forme est brièvement décrite en annexe II.





## Chapitre 9

# Systèmes embarqués et méthodologie AAA

Les travaux présentés dans ce chapitre s'inscrivent dans le cadre de recherches conjointes menées avec l'équipe Architecture de notre laboratoire. Une collaboration active s'est en particulier engagée avec Mickaël Raulet (doctorant), qui s'intéresse aux problématiques de prototypage rapide d'applications de traitement du signal sur architectures distribuées hétérogènes embarquées. L'article [RBD<sup>+</sup>03] résulte de ce travail effectué en commun.

La **méthodologie AAA** (Adéquation Algorithme Architecture) s'est imposée afin de répondre à la préoccupation suivante : développer des méthodes systématiques de meilleure mise en correspondance entre un algorithme d'une part, et une architecture d'autre part. Pour ce faire, l'outil SynDEx réalise la phase d'adéquation et de génération automatique de code, sur un critère de minimisation du temps d'exécution. SynDEx repose sur des formalismes de graphes, à la fois pour spécifier le problème d'adéquation (terme dont nous donnerons la définition en section 1), et pour le résoudre. La référence [Déf04] résume l'ensemble des recherches et des expérimentations menées au sein du laboratoire. Nous nous appliquerons donc à montrer tout particulièrement ici les derniers développements qui concernent les décompositions pyramidales du LAR (chapitres 5, 6 et 7) ainsi que le codeur vidéo LAR, et leur implantation sur des architectures hétérogènes multi-composants.

Le portage complet des méthodes LAR (LAR simple, LAR-APP, Interleaved S+P, vidéo) sous SynDEx vise à fournir les éléments suivants :

- *validation fonctionnelle* de l'application considérée (application mono-PC),
- *applications de test* pour le prototypage rapide sur les architectures hétérogènes embarquées disponibles au laboratoire,
- *plate-formes de démonstration*,
- *description synthétique* du LAR, à destination des utilisateurs de la méthode, grâce notamment à la visualisation de l'algorithme sous forme de graphes flot de données.

Ce chapitre s'articule autour de quatre paragraphes principaux à savoir : présentation générale des concepts inhérents à SynDEx et à la méthodologie AAA (§1), puis définition des éléments de description intégrés dans SynDEx (§2), modélisation des codecs LAR avec les résultats associés sur cible mono-PC (section 3), et, enfin, résultats des expérimentations menées sur cibles multi-composants (section 4).

# SynDEx et méthodologie AAA

## 1.1 Présentation générale

SynDEx <sup>1</sup> est un outil universitaire et gratuitement téléchargeable, dont la signification est "Synchronized Distributed Execution". Il s'inscrit dans le cadre du projet OSTREA de l'INRIA Rocquencourt, sous la direction d'Yves Sorel. Plus précisément, l'outil est le fruit des recherches menées sur la méthodologie AAA [LS92], pour le **prototypage rapide et l'implantation optimisée d'applications distribuées temps réel embarquées** [Déf04]. La méthodologie a pour objectif de résoudre le problème d'optimisation suivant : choisir une implantation sur une architecture matérielle donnée, dont les performances, déduites des caractéristiques des composants considérés, respectent les contraintes temps-réel et "d'embarquabilité". Les architectures cibles peuvent être aussi bien un ensemble de processeurs, qu'un ensemble de composants spécifiques. Les grands principes d'utilisation de l'outil sont résumés sur la figure 9.1.

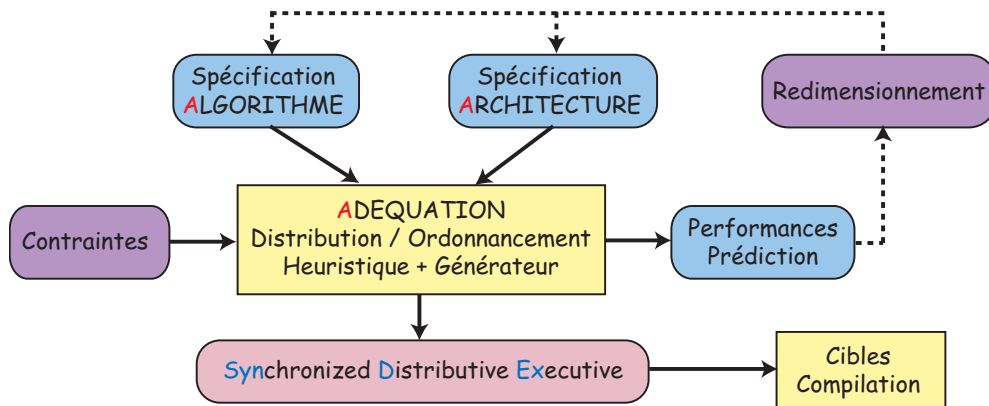


FIG. 9.1: Principe général de SynDEx

## 1.2 L'adéquation

SynDEx repose sur des modèles de graphes, autant pour la spécification de l'algorithme et de l'architecture, que pour la résolution du problème d'adéquation à travers les transformations de ces graphes. La figure 9.2 illustre l'environnement SynDEx, avec les graphes de description d'une part, et le graphe temporel résultant de leur transformation et conduisant au partitionnement d'autre part.

**Graphe d'algorithme.** L'algorithme est décrit sous la forme d'un "graphe de dépendances de données conditionné factorisé" [LS97] : c'est un hypergraphe orienté acyclique (DAG), dont les sommets s'apparentent à des opérations partiellement ordonnées à travers leurs dépendances de données représentées par des hyperarcs [Pra86]. L'outil autorise à la fois la **hiérarchie** et le conditionnement : chaque opération du graphe correspond ainsi soit à une **opération atomique**, soit à un sous-graphe dont l'exécution peut être conditionnée. La répétition implicite ou explicite de sous-graphes est également supportée. Un tel formalisme permet la description d'applications complexes, où le graphe d'algorithme exprime le **parallélisme potentiel** de l'application.

<sup>1</sup>[www.syndex.org/](http://www.syndex.org/)

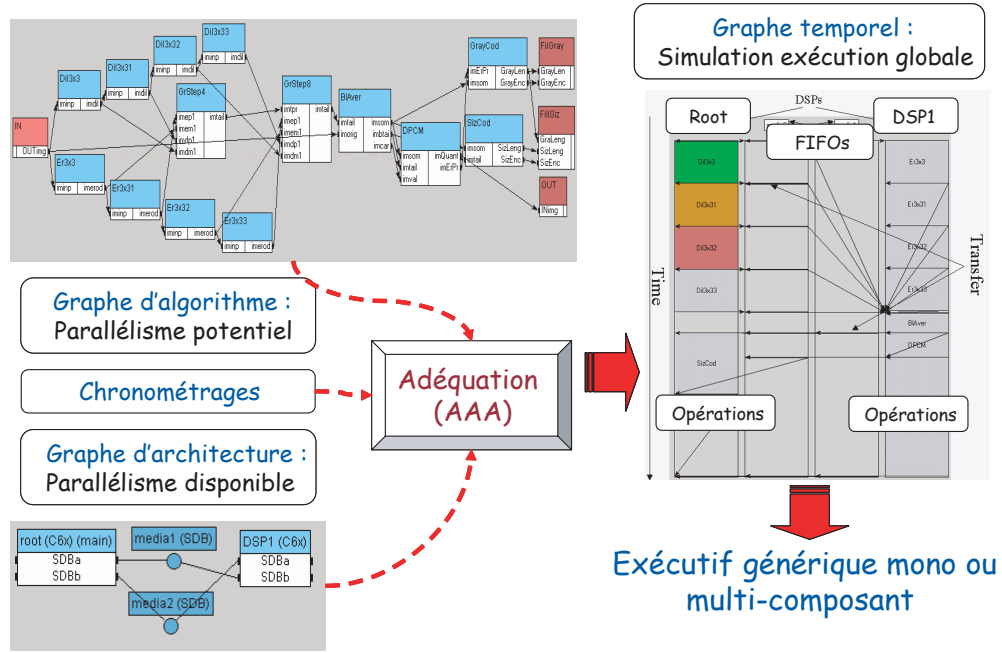


FIG. 9.2: Environnement SynDEx

**Graphe d'architecture.** Le graphe d'architecture est un graphe orienté modélisant le **parallélisme disponible** d'une cible multicomposants hétérogène. Une architecture multi-processeurs se représente à un haut niveau par des unités de calcul reliées par des média de communication. On dénombre cinq types de sommets [Gra00] :

- l'opérateur ou le séquenceur d'instructions pour séquencer des opérations de calcul,
- le communicateur ou le séquenceur de communication pour séquencer des opérations de communication,
- le bus avec ou sans arbitre pour diffuser les données,
- la mémoire  $RAM^2$  à accès aléatoire,
- la mémoire  $SAM^3$  à accès séquentiel.

Un médium de communication est représenté au minimum sous la forme de bus et d'une mémoire, de type RAM ou SAM. Il peut de plus intégrer des communicateurs lorsque la communication s'effectue parallèlement au traitement dans un processeur (présence matérielle de DMA<sup>4</sup>). Le modèle d'architecture adopté dans SynDEx reste suffisamment générique pour que l'architecture interne globale de n'importe quel processeur puisse être décrite.

**Implantation.** L'implantation est la phase du processus consistant à distribuer et ordonnancer à la fois les opérations de l'algorithme et les opérations de communication qui en découlent, sur les ressources matérielles de calculs (processeur) et de communication (média). Pour évaluer la charge de l'application, il est nécessaire de connaître les temps de traitement (propres aux processeurs) associés à chaque opération atomique. Les temps de communication se déduisent eux à partir d'un temps unitaire par type de données transmises. L'ordonnancement consiste à rendre total l'ordre partiel associé à chaque sous-graphe de l'algorithme distribué sur un processeur, tout en respectant

<sup>2</sup>RAM : Random Access Memory

<sup>3</sup>SAM : Sequential Access Memory

<sup>4</sup>DMA : Direct Memory Access

les dépendances de données allouées aux média. L'ordonnancement est ici non préemptif et hors ligne (fixé a priori). Le résultat d'implantation optimisé par SynDEx se modélise sous la forme d'un nouveau graphe, obtenu par transformation des graphes d'architecture et d'algorithme.

**Remarque.** SynDEx fournit une visualisation des résultats temporels prédits de l'implantation, sous la forme d'un graphe temporel. Cette prédiction permet de vérifier le parallélisme obtenu et, si nécessaire, d'optimiser l'implantation. Pour cela, l'utilisateur interagit avec l'heuristique, en adaptant la granularité des opérations de l'algorithme, ou en modifiant l'architecture.

### 1.3 Génération de code automatique

La génération automatique d'exécutifs se fait suivant des règles décrivant la transformation d'un graphe d'implantation optimisé en un graphe d'exécution. Dans ce paragraphe, les notions de graphes synchronisés, de macro-code et de code compilable sont successivement abordées.

**Graphes synchronisés.** Le terme "synchronisé" présent dans l'acronyme SynDEx indique que la cohérence de l'exécution de l'ensemble de l'application est assurée par la synchronisation des différentes machines séquentielles, à l'aide en particulier de mécanismes de sémaphores.

**Macro-code.** SynDEx génère un "macro-code" fondé sur le formalisme précédent, indépendant des langages utilisés par les processeurs. L'exécutif créé pour chaque processeur, contient les allocations de mémoire, les initialisations, et la séquence ordonnée d'opérations sur chaque opérateur et chaque communicateur codé sous la forme d'une boucle infinie. Le macro-code est composé de macro-instructions, regroupées selon les deux catégories suivantes :

- *les macros applicatives*, réalisant les opérations de l'algorithme. Chaque macro-instruction est donc un appel de fonction, qui devra être compilée en fonction du langage utilisé (C, C++, assembleur, ...),
- *les macros systèmes* qui définissent le noyau d'exécutif, et supportent les opérations telles que la gestion de la mémoire, le séquençement, les communications et les synchronisations. Ces macros sont donc dépendantes du processeur et/ou des média de communication.

**Code compilable.** Afin d'obtenir un code compilable pour chaque processeur, le code générique doit être traduit en un code spécifique à un langage et une architecture donnés. Pour ce faire, SynDEx s'appuie sur le macro-processeur GNU M4 [Sei00].

Un macro-processeur se définit comme un programme capable d'interpréter des commandes (macros) elles-mêmes définies par l'utilisateur. Il existe donc un dictionnaire qui associe chaque nom de macro avec une définition de substitution. Dans un souci de réutilisation et de simplicité, ce dictionnaire se scinde hiérarchiquement en plusieurs bibliothèques portant chacune l'extension ".m4x", où trois grandes catégories se distinguent :

- *bibliothèque générique* : c'est la bibliothèque SynDEx.m4x, lue en premier, contenant un dictionnaire de nouvelles macro-instructions de lectures pour GNU M4,
- *bibliothèque dépendante de l'application* : elle permet de définir les appels des fonctions réelles à appeler (dépendantes du langage) et ses paramètres formels,

- *bibliothèque dépendante de l'architecture* : il s'agit principalement de dictionnaires propres à un type de processeur ou à un médium de communication.

Développer un "noyau SynDEx" pour une plate-forme cible donnée, revient par conséquent à construire des bibliothèques de macros servant d'interface avec le code générique, et à éventuellement réaliser les primitives de base nécessaires (synchronisation et communication entre autres).

La figure 9.3 récapitule l'ensemble des opérations réalisées entre description des graphes sous SynDEx, et implantation sur cible mono ou multicomposants. La dernière phase de compilation et de chargement est dépendante du type de processeur. Dans le laboratoire, où l'on utilise majoritairement des DSP<sup>5</sup> de Texas Instrument en tant que processeurs embarqués, Code Composer tient lieu d'environnement de développement.

## 2 Les éléments de description sous SynDEx

SynDEx appartient à la famille des logiciels de CAO, où l'application visée est décrite sous la forme d'un **graphe flot de données**. La modélisation des codeurs LAR nécessite la connaissance préalable des différents éléments implantés sous SynDEx et présentés dans cette section.

### 2.1 Graphe d'algorithme

**Considérations générales.** Les sommets du graphe caractérisent une action ("Operation"), une entrée/sortie ("Sensor" ou "Actuator"), une constante ("Constant") ou une mémorisation de données ("Memory"). Ces sommets sont reliés par des arcs qui représentent un transfert de données. Le nombre de sommets rencontrés lors du parcours du graphe d'algorithme indique la **granularité** de la description de l'application.

Dans notre cas, chaque opération élémentaire correspond à une fonction écrite en langage C : à sa définition sous SynDEx, nous devons déterminer l'ensemble des paramètres passés sous forme de buffers ("ports") de tailles fixes (allocation statique). Il est alors possible d'instancier autant de fois que nécessaire la même opération dans une application donnée (appels successifs à la fonction paramétrable).

**Hiérarchie.** Chaque opération du graphe peut être à son tour décrite par un sous-graphe permettant une spécification **hiérarchique** de l'algorithme, jusqu'aux "opérations atomiques" que l'on ne peut spécifier à l'aide d'un sous-graphe. Cette spécificité de SynDEx autorise une certaine souplesse dans la définition de l'application, et permet ainsi d'adapter la granularité du graphe flot de données. Il est à noter que le logiciel ne supporte pas la notion de paramètre unique définissant une entrée/sortie : chaque port correspond soit à une entrée, soit à une sortie. Une variable d'entrée/sortie doit ainsi être décrite par deux ports distincts.

**Itération et répétition.** L'itération correspond à une répétition temporelle d'une même opération, caractérisée par une dépendance de données. La répétition est à prendre ici au sens

---

<sup>5</sup>DSP : Digital Signal Processor

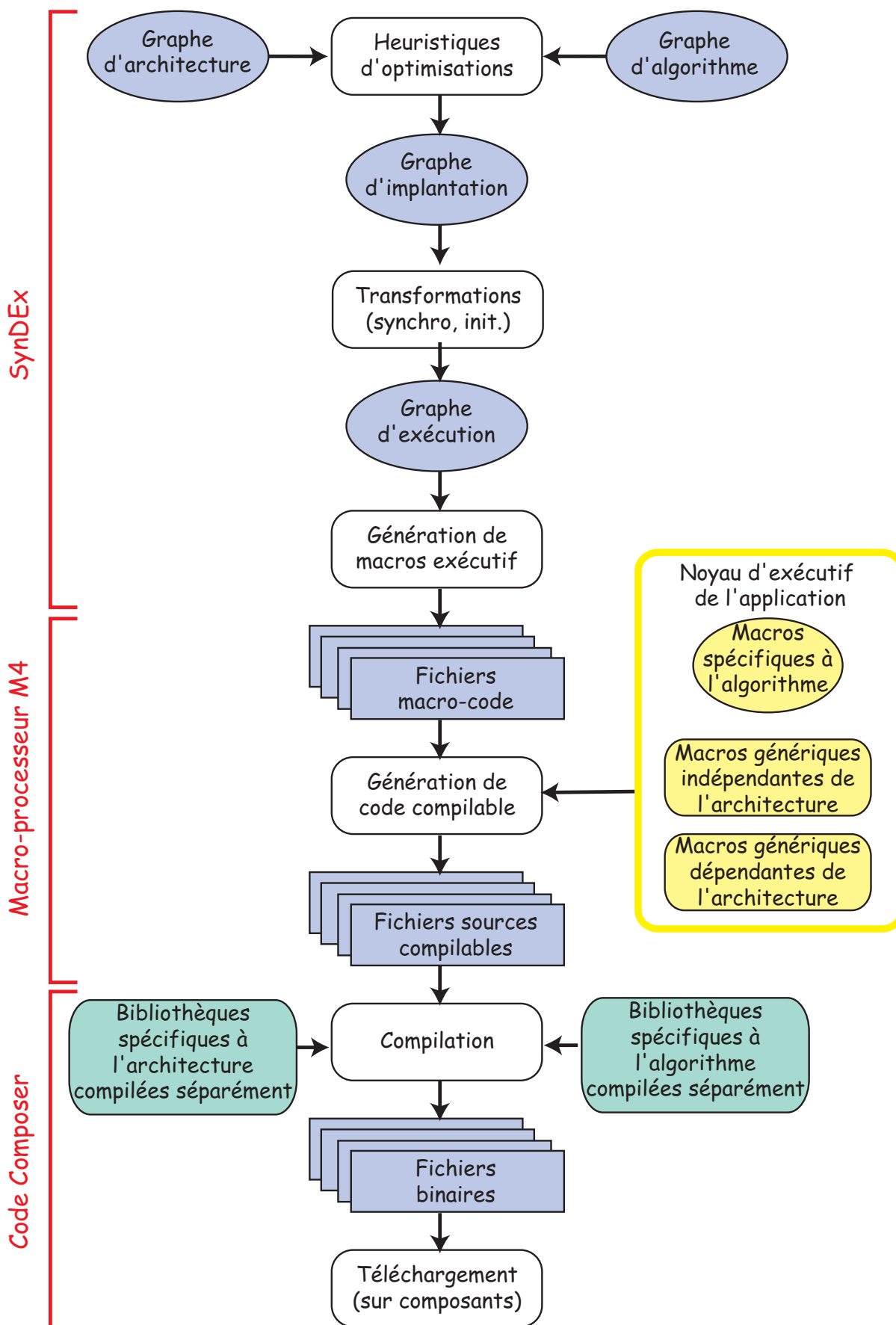


FIG. 9.3: Processus complet de prototypage

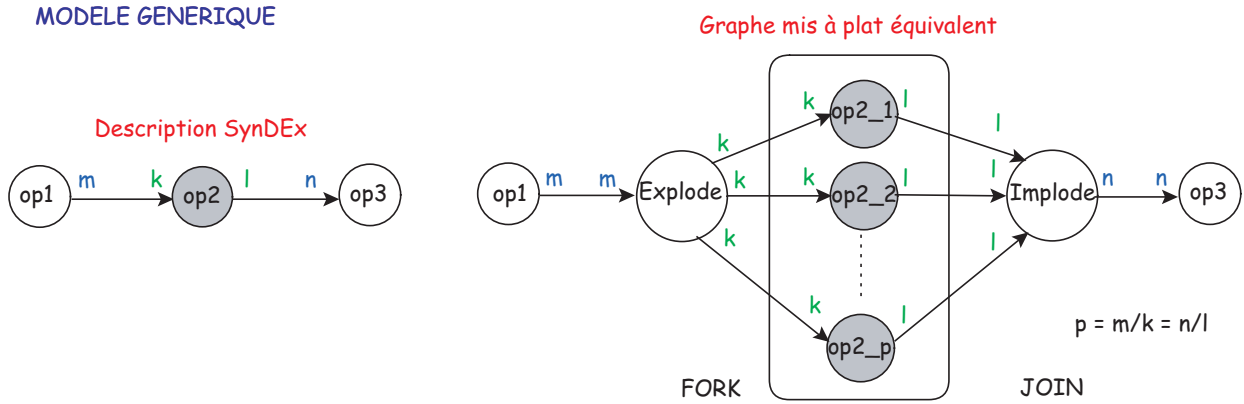


FIG. 9.4: Répétition implicite d'une opération (mécanisme de Fork / Join)

de SynDEx : il s'agit de fait d'un appel répété à la fonction, sans dépendance de données, avec une répartition spatiale des buffers d'entrées et de sorties. Ce type de structure fait apparaître un parallélisme potentiel supplémentaire.

La répétition implicite (temporelle ou spatiale) consiste à spécifier la taille d'un tableau connecté à une entrée (resp. sortie) de l'opération considérée, comme étant un sous-multiple de la taille de cette entrée (resp. sortie) : c'est le mécanisme Fork-Joint, illustré par la figure 9.4 [RBD<sup>+</sup>03]. Dans le cas d'une itération, il est également possible de déterminer de façon explicite le nombre de répétitions nécessaires à l'application.

## 2.2 Graphe d'architecture

La description des architectures cibles a été plus particulièrement réalisée par Mickaël Raulet. Elle fait apparaître l'ensemble des composants mis en œuvre, ainsi que tous les média de communication utilisés. Nous ne traiterons pas ici des méthodes de développement qui ont précédées l'élaboration des bibliothèques nécessaires à la construction du graphe d'architecture : le lecteur pourra se reporter à l'article [RUN<sup>+</sup>05].

## 2.3 Portage de l'application sur architecture hétérogène

Le processus complet de prototypage sur architecture hétérogène multi-processeurs, comporte trois phases.

- *Phase 1* : le **graphe d'application** (graphe d'algorithme associé au graphe d'architecture ne comprenant qu'un PC) est tout d'abord décrit sous SynDEx. La génération automatique de code fournit un programme en C pour une première implantation sur PC. L'utilisateur crée les fonctions en langage C associées à chacune des opérations du graphe d'application et teste les fonctionnalités de son application avec un environnement de compilation standard.
- *Phase 2* : le graphe d'application est ensuite directement utilisé pour des **implantations mono-processeur avec chronométrages**, pour chaque type de processeur cible. Le code généré insère automatiquement les fonctions de chronométrage de chaque fonction rencontrée. L'utilisateur compile alors les sources et lance l'exécution du programme avec l'outil



adéquat à la cible. Les temps de chacune des fonctions (opérations du graphe) s'affichent en fin d'exécution. Cette étape est l'occasion de tester diverses optimisations relatives aux cibles (optimisation de l'écriture du programme, utilisation du langage assembleur ou de bibliothèques optimisées livrées avec le composant). Les différents temps obtenus, après optimisations éventuelles, peuvent alors être reportés sous SynDEx.

- *Phase 3* : SynDEx réalise l'**adéquation et un exécutif distribué optimisé**. Plusieurs configurations architecturales peuvent être simulées en faisant varier le nombre et le type des processeurs ou connexions.

### 3 Modélisation du codec LAR

Le portage automatique d'une application sur une architecture cible nécessite la modélisation préalable de son algorithme sous forme de graphes flots de données, via le logiciel SynDEx. Cette étape impose le respect de certaines contraintes de développement (§3.1). Si les tous premiers travaux ont consisté en la description du codec LAR simple (chapitre 1), nous abordons dans cette section la modélisation des approches pyramidales LAR-APP et Interleaved S+P, ainsi que celle de l'extension du LAR à la video (§3.2). Enfin, est exposée au paragraphe 3.3 la validation fonctionnelle des applications ainsi décrites sur un système mono-PC.

#### 3.1 Objectifs du travail effectué - Contraintes de développement

Rappelons les objectifs que nous nous sommes fixés : validation fonctionnelle, réalisation d'applications destinées au prototypage rapide et de plate-formes de démonstration, et description synthétique à caractère pédagogique du LAR. Pour ce faire, il nous a fallu dans un premier temps transcrire notre algorithme de compression sous forme de graphes flot de données, implantés sous SynDEx.

Une part importante du travail consiste à écrire les fichiers ".m4x" permettant la génération d'un code compilable sous Visual C++ et sous Dev C++. L'utilisation de SynDEx comme générateur automatique de code (fonction principale de l'application) induit également certaines contraintes de développement. La première concerne la définition des paramètres d'entrée/sortie d'une fonction écrite en langage C et associée à une opération atomique du graphe flot de donnée. L'ordre de ces paramètres doit être en totale coïncidence avec celui donné dans le logiciel de CAO. D'autre part, SynDEx n'autorise pas d'allocation dynamique des variables : l'occupation mémoire doit être déterminée au préalable. SynDEx effectue alors automatiquement la réservation statique des paramètres nécessaires. Enfin, lors du développement même des fonctions, nous nous efforçons d'en extraire le parallélisme potentiel.

#### 3.2 Description du codage source sous SynDEx

La description des algorithmes relatifs aux méthodes LAR-APP, Interleaved S+P, et au LAR adapté à la vidéo conduisent à la définition de nombreuses opérations atomiques sous SynDEx. L'objectif n'est pas ici de les détailler entièrement, mais plutôt de comprendre la méthodologie de

conception de l'application utilisée. Le décodeur étant en parfaite symétrie avec le codeur, nous ne montrerons ici que la description SynDEx des codeurs analysés. Il est à noter que les opérations globales du LAR-APP et de l'Interleaved S+P sont semblables : seul le code interne des fonctions de décomposition pyramidale diffère.

**Niveau le plus haut de la hiérarchie SynDEx.** Au niveau hiérarchique le plus haut, se trouvent les fonctions de récupération des paramètres du codec, de l'image à traiter et des opérations globales de codage et de décodage. SynDEx itère le graphe d'algorithme à l'infini : ceci implique que la majorité des opérations du graphe sont exécutées (de façon ordonnées) une infinité de fois. Certaines fonctions ne doivent cependant n'être appelées qu'une seule fois (typiquement la récupération des paramètres du codec) : il suffit alors de le spécifier dans le fichier ".m4x" à destination du macro-processeur M4. La figure 9.5 montre la description globale réalisée pour le codec LAR-APP, ainsi que la représentation graphique des éléments élémentaires évoqués au paragraphe 2.

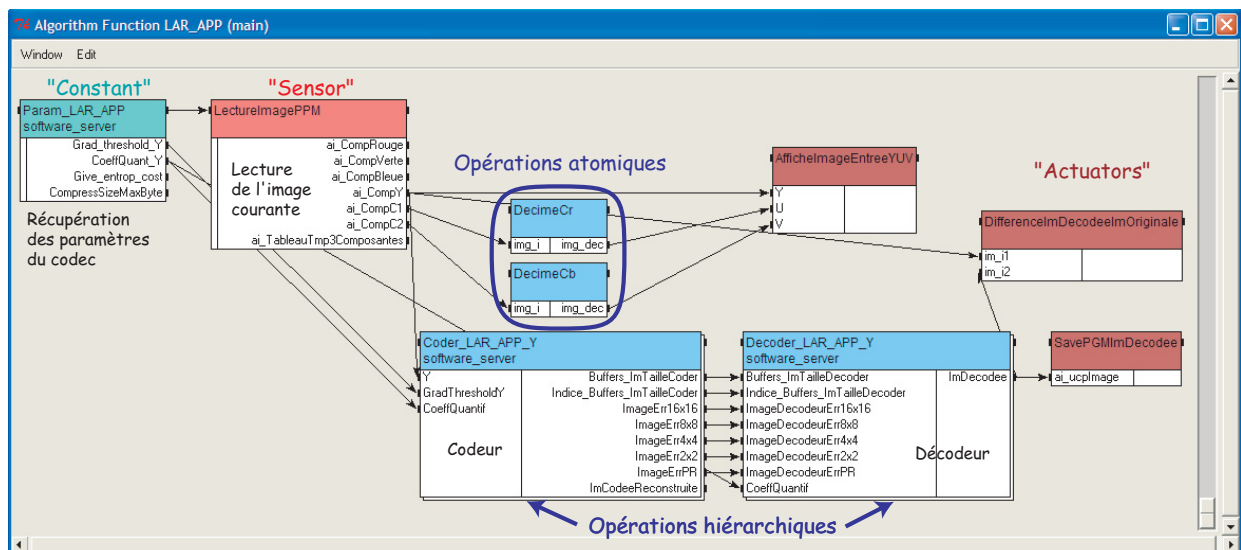


FIG. 9.5: Graphe correspondant au niveau le plus haut de la hiérarchie SynDEx (Interleaved S+P et LAR-APP). Visualisation des éléments de description élémentaire d'une application.

Notons qu'il est possible d'ouvrir et sauver indifféremment des images au format PGM ou PPM, de lire des vidéos au format YUV ou Y, ou encore d'exploiter le flux issu d'une webcam. En effet, afin de tenir compte des spécificités du traitement des images d'un point de vue entrée/sortie, ont été créées puis encapsulées des fonctions d'acquisition d'images (par webcam ou par lecture de fichier), et d'affichage (une fenêtre associée à chaque appel de fonction), sous environnement standard de développement (Visual C++ et Dev C++). L'utilisateur est ainsi en mesure d'instancier directement des opérations spécifiques de visualisation dans son graphe d'algorithme. Si le graphe d'architecture comporte un ou plusieurs PC, l'exécutif généré fera apparaître automatiquement les séquences désirées par le développeur dans des fenêtres de visualisation.

**Étapes majeures de la méthode de codage pyramidal.** La figure 9.6 indique les grandes phases de l'Interleaved S+P : construction de la grille, construction et descente de la pyramide, mise en forme des flux à transmettre. Nous visualisons de plus les fonctions annexes qui traitent du post-traitement, du calcul de l'entropie d'ordre zéro, de sauvegarde et d'affichage des images produites.

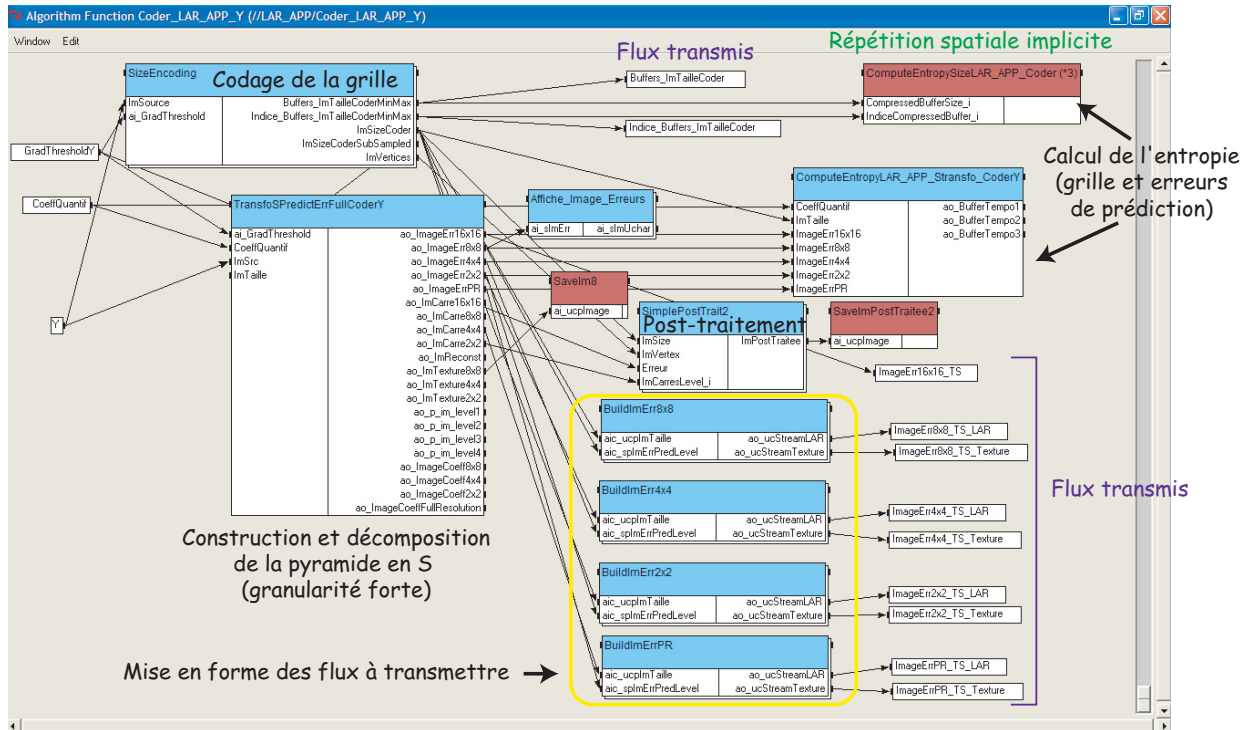


FIG. 9.6: Description commune à l'Interleaved S+P et au LAR-APP.

**Sous-échantillonnage non uniforme : la grille LAR.** La première opération appelée lors du processus de codage d'une image correspond à la définition de la grille et à la compression de cette information, et est illustrée par la figure 9.7. Deux représentations possibles d'une répétition temporelle sont données :

- **mise à plat** du graphe de calcul du partitionnement non uniforme de l'image originale, expression directe du parallélisme potentiel de l'opération concernée,
- **répétition implicite** de la fonction destinée à la compression des données codant la grille.

**Approche pyramidale prédictive.** L'ensemble des méthodes pyramidales proposées dans ce mémoire font appel à la même structure de composition et décomposition pyramidale. La figure 9.8 permet de visualiser l'enchaînement des étapes nécessaires à la compression d'une image donnée par le LAR-APP ou l'Interleaved S+P<sup>6</sup>. La construction de la pyramide précède comme de juste l'ensemble des opérations de décomposition. Ainsi sont distinguées les deux descentes de la pyramide (obtention de l'image LAR basse résolution dans un premier temps, la texture ensuite), de même que les deux passes prédictives internes à chaque niveau de décomposition. Le codage de

<sup>6</sup>L'Interleaved S+P et le LAR-APP possèdent la même structure globale : la distinction est essentiellement opérée pour chaque étape élémentaire de la décomposition pyramidale (fonctions "ConstructionLeveli" et "TextureLeveli" de la figure 9.8).

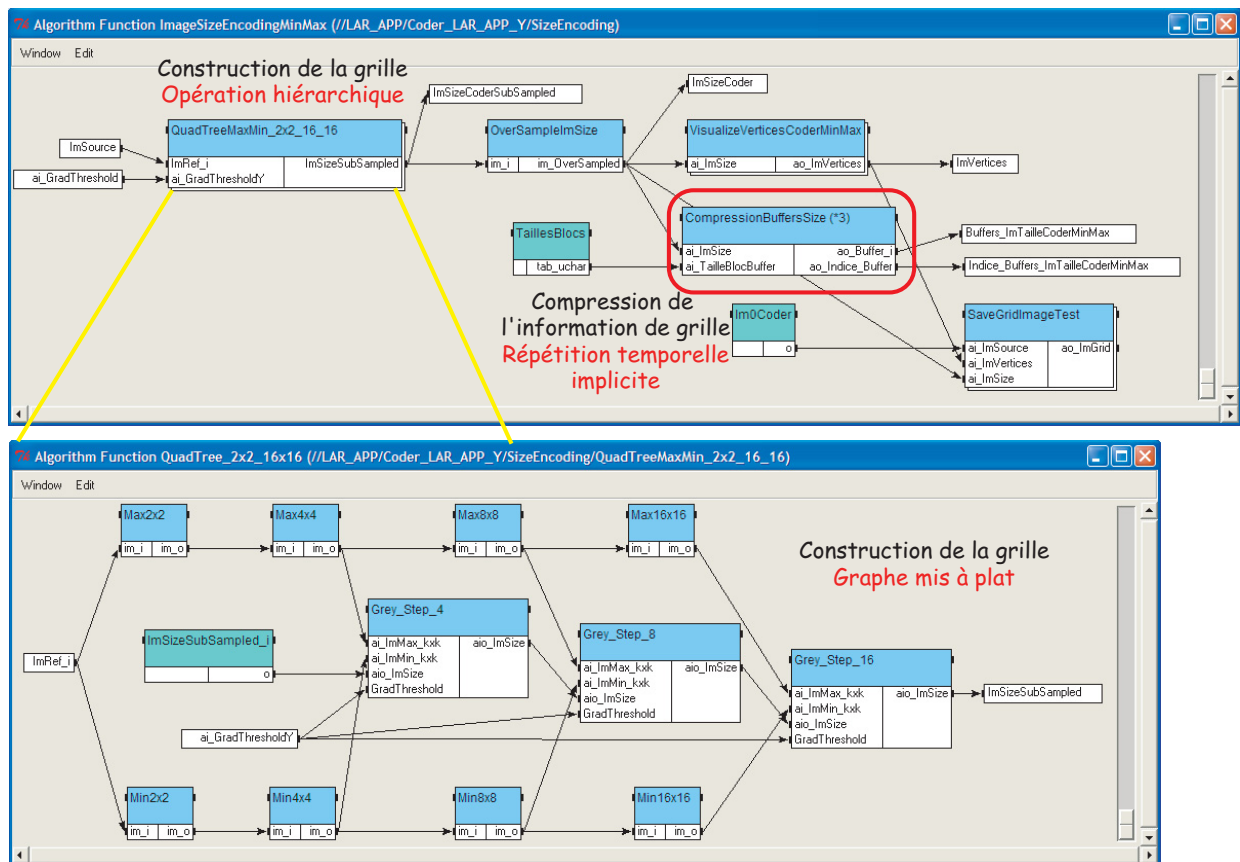


FIG. 9.7: Construction et compression de la grille.

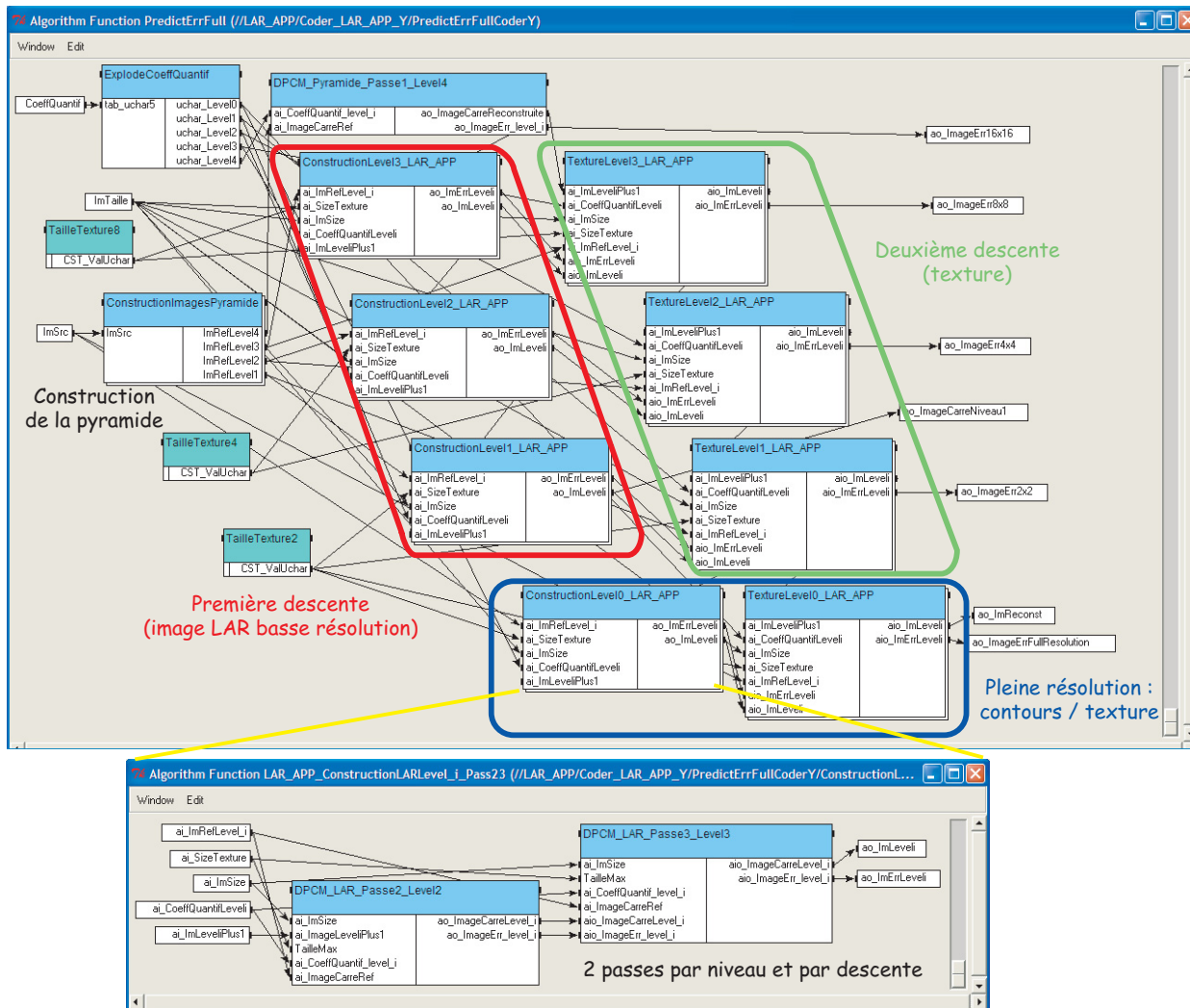


FIG. 9.8: Visualisation des différentes étapes de la décomposition pyramidale.

l'information en pleine résolution est réalisé en deux étapes successives : traitement des contours (blocs de taille minimale) puis traitement des zones de texture (autres blocs).

**Video et segmentation.** Le codec vidéo du LAR repose sur l'exploitation de la carte de segmentation automatiquement disponible à la fois au codeur et au décodeur. La figure 9.9 présente la description SynDEx du codeur correspondant (images "Intra" seulement). La grille est cette fois-ci déterminée à partir de la donnée des 3 composantes (luminance et chrominances partagent le même partitionnement). Un schéma MICD, utilisant le prédicteur médian, encode l'information de valeur des blocs de luminance. La segmentation est ensuite mise en place : les bits de contrôle de la segmentation par la chrominance constituent la seule information transmise au décodeur pour cette phase. Les composantes de chrominance sont alors codées au niveau région (prédiction médian). Il est à noter que le codeur entropique Golomb-Rice adapté à notre problématique est ici implanté [Déf04].

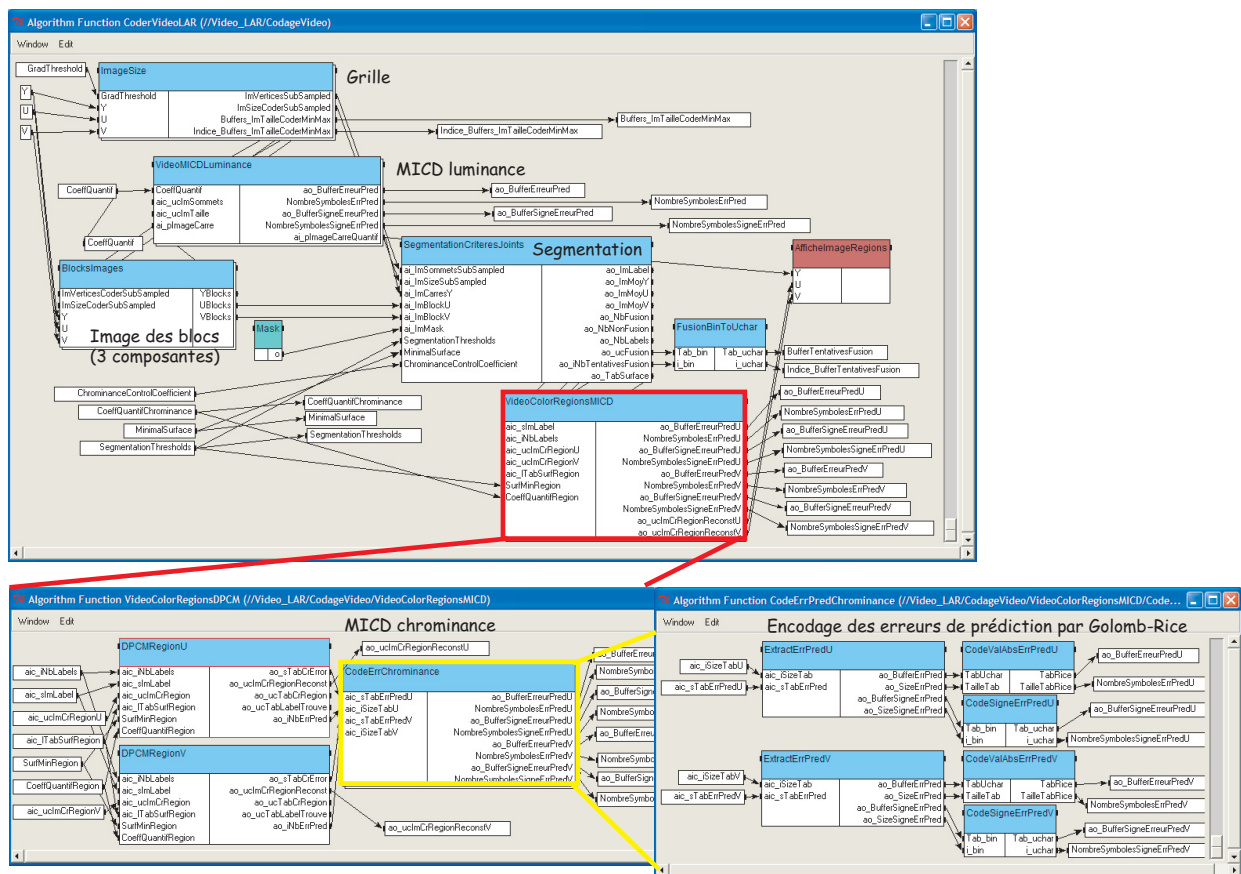


FIG. 9.9: Application vidéo du LAR et segmentation.

### 3.3 Validation fonctionnelle sur un seul PC

Partant directement du graphe d'algorithme sous SynDEx, une première implantation mono-processeur sur PC, sous environnement standard, est générée pour le test fonctionnel. La figure 9.10 présente le graphe d'architecture correspondant. Le LAR simple, le LAR-APP, l'Interleaved



**S+P et les premiers développements réalisés pour l'application vidéo associée à la segmentation ont donc été vérifiés et validés.**

L'application ainsi produite fonctionne en mode sans débogage et autorise ainsi les vérifications au fur et à mesure des optimisations faites par le développeur jusqu'au prototype final.

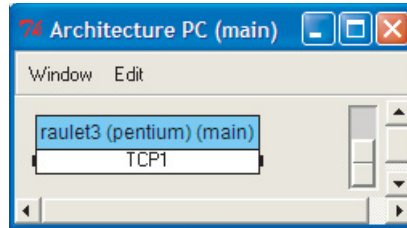


FIG. 9.10: Graphe d'architecture pour application mono-PC (pentium).

## 4 Environnement multi-processeur

Dans cette section sont présentées les différentes architectures sur lesquelles nos algorithmes, et en particulier l'Interleaved S+P ont été implantés. Nous avons privilégié les résultats obtenus sur des cibles multi-PC ainsi que sur les cibles mixtes multi-PC, multi-DSP. La généricité de SynDEx permet de passer aisément d'une plate-forme à une autre, par la seule modification de l'algorithme d'architecture.

### 4.1 Application client-serveur : cible multi-PC

Le noyau TCP définit les macros nécessaires à la mise en œuvre de transmissions sur liens TCP/IP. Ainsi, il est possible de prendre en compte des cartes multi-DSP connectées par un lien TCP à un PC hôte, ou encore des environnements multi-PC. Grâce à ce medium de communication, des applications de type client/serveur sont aisément réalisées. Le graphe d'algorithme correspondant est illustré par la figure 9.11.

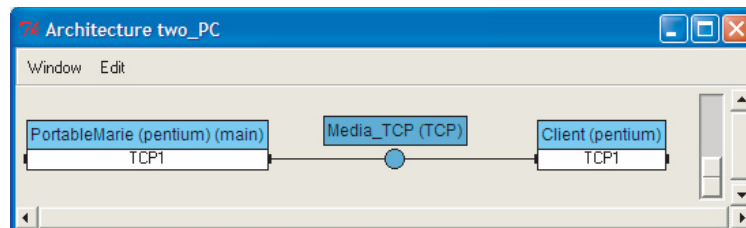


FIG. 9.11: Graphe d'architecture pour application client-serveur.

Après génération automatique du code distribué sur les deux PC (Pentium 4, 2.8 GHz), nous avons pu vérifier et valider le comportement des méthodes pyramidales sans perte LAR-APP et Interleaved S+P. Le codage est alors affecté au processeur "serveur", et le décodage est implanté sur le processeur "client".

A partir des chronométrages réalisés sur les fonctions principales de l'Interleaved S+P, nous obtenons le tableau 9.1. Cette version de l'algorithme fait intervenir le codage entropique Golomb-Rice (parties "Encodage/Décodage des erreurs par niveau de la pyramide"), afin d'expérimenter une chaîne complète de compression d'une image sans perte. Le codage de notre représentation en taille de blocs variable correspond à la méthode décrite au chapitre 8, paragraphe 3.1.1.

Chronométrages des fonctions ( $\mu s$ )			
CODAGE		DECODAGE	
Calcul du Quadtree	1 904		
Codage de la grille	175	Décodage de la grille	1 146
Traitement pyramidal	16 367	Traitement pyramidal	10 447
<i>Encodage des erreurs par niveau</i>		<i>Décodage des erreurs par niveau</i>	
Niveau $l = l_{max} = 4$	87	Niveau $l = l_{max} = 4$	51
Niveau $l = 3$	278	Niveau $l = 3$	145
Niveau $l = 2$	1 070	Niveau $l = 2$	483
Niveau $l = 1$	4 232	Niveau $l = 1$	1 604
Niveau $l = 0$	16 894	Niveau $l = 0$	5 509
Codage (total)	41 804	Décodage (total)	19 504

TAB. 9.1: Relevé des temps d'exécution des principales fonctions de l'Interleaved S+P sur un PC de type Pentium 4, 2.8 GHz, pour la première image de la séquence "foreman\_cif.yuv" ( $352 \times 288$ ).

Si l'on applique la **contrainte temps-réel** de la vidéo, celle-ci impose que le processus de codage ou de décodage d'une image complète soit effectué en moins de 40 ms. Si le décodage s'avère aisément temps-réel (19.5 ms), la phase de codage atteint les 41.8 ms et dépasse la consigne.

Analysons maintenant les différents temps observés. La première constatation simple est la suivante : le calcul de la partition Quadtree n'est réalisé qu'au codage, et occupe 1.9 ms. L'étape d'encodage de la grille se révèle performante en termes de temps d'exécution. Au décodeur, l'opération inverse visant à reconstruire la grille de représentation ne prend quant à elle que 1.14 ms.

**Traitement pyramidal.** Les fonctions les plus lourdes concernent bien évidemment le traitement pyramidal par prédiction. Le décodeur étant symétrique au codeur, les temps d'exécution de la décomposition doivent être équivalents. Comment expliquer la nette différence de temps observée ? Au codeur, la fonction étudiée dédiée au traitement pyramidal de l'image inclut les opérations successives de construction pyramidale : celle-ci ne prenant que  $92\mu s$ , la solution réside ailleurs. Nous avons donc étudié au plus près l'ensemble des fonctions appelées dans la décomposition pyramidale. De façon à ce que notre code soit le plus générique possible, une phase de quantification/déquantification et récupération de l'erreur quantifiée est implantée. Il apparaît clairement que cette étape n'est absolument pas optimisée. En particulier, les opérations de quantification/déquantification sont implantées au moyen de simples divisions : bien évidemment, leur implantation par décalages successifs s'avérerait largement profitable !

Dans l'optique d'une implantation à objectif unique, à savoir la réalisation d'un codage sans perte, il est possible d'entièrement supprimer la phase de quantification. De nouveaux chronométrages ont donc été réalisés (tableau 9.2). Le gain s'avère conséquent, en particulier au codage (10 ms au lieu de 16 ms). Cette simple modification permet d'obtenir un schéma de codage à son tour temps réel !



Chronométrages des fonctions ( $\mu s$ )			
CODAGE		DECODAGE	
Traitement pyramidal	10 265	Traitement pyramidal	4 071

TAB. 9.2: Relevé des temps d'exécution du traitement pyramidal après suppression des phases de (dé)quantification sur un PC de type Pentium 4, 2.8 GHz, pour la première image de la séquence "foreman\_cif.yuv" ( $352 \times 288$ ).

**Encodage des erreurs de prédiction.** L'encodage des erreurs par niveau de pyramide implique tout d'abord le **regroupement des coefficients** à transmettre selon leur nature, tout en différenciant les données codant l'image LAR basse résolution de l'information de texture (voir §5.2 du chapitre 6). Sur chacun des sous-flux ainsi construits, nous appliquons le **codage de Golomb-Rice** tel que décrit dans [Déf04]. Nous pouvons visualiser les différentes phases du codage sur la figure 9.12.

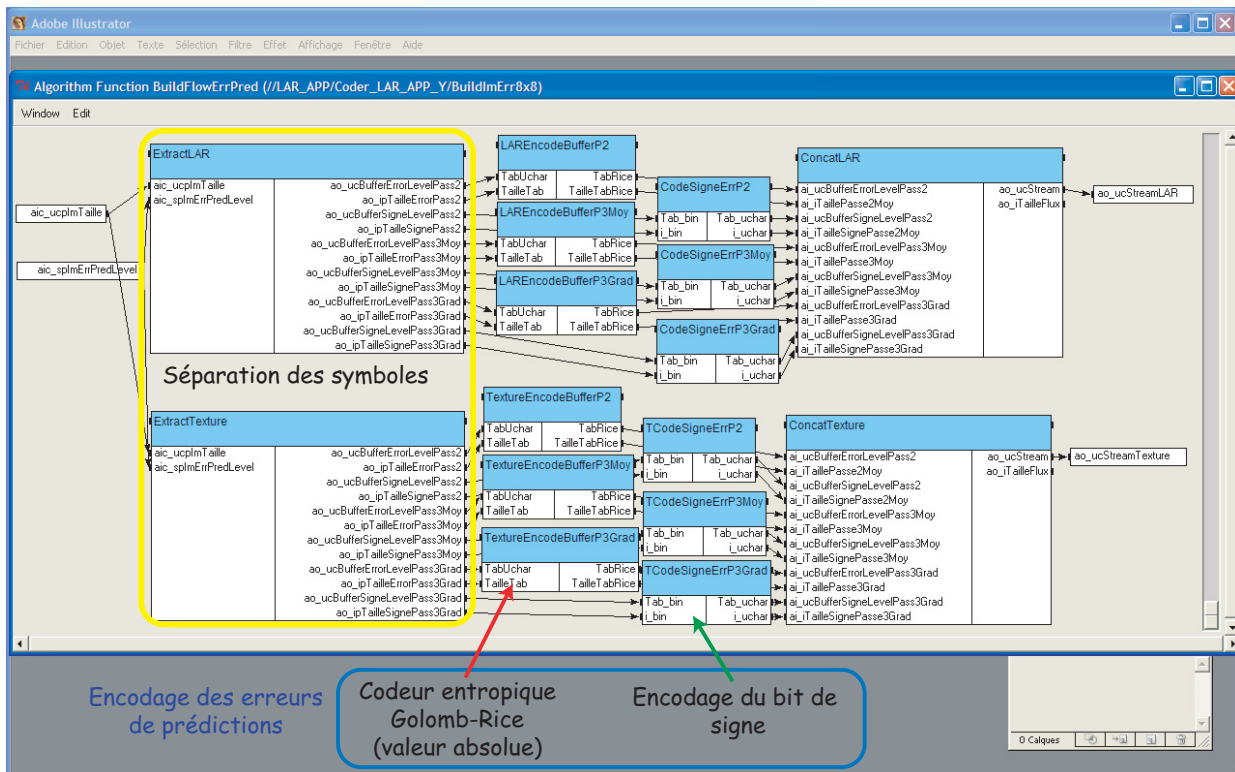


FIG. 9.12: Description SynDEx de l'encodage des erreurs de prédiction et de la mise en forme des flux pour l'Interleaved S+P.

La première étape consiste à séparer le bit de signe de la valeur absolue, de telle sorte que seule cette dernière soit encodée par les codes de Golomb-Rice. Ces codes se définissent de la façon suivante. Soit une valeur  $n$ , codée sur 8 bits. On fixe alors  $k$  le nombre optimal de bits de poids faible de  $n$  (au sens de la longueur moyenne de code engendrée). Le code construit est alors constitué d'une suite de 1 de longueur  $\lfloor n/2^k \rfloor$ , d'un 0 délimiteur et des  $k$  bits de poids faibles, conduisant à une longueur totale de code égale à  $\lfloor n/2^k \rfloor + 1 + k$ . Une phase préalable d'estimation de  $k$  se révèle donc indispensable au codeur (la valeur de  $k$  étant transmise au décodeur, celui-ci ne doit pas réitérer cette opération). De nouveau, les calculs effectués au décodeur sont symétriques

à ceux exécutés au codeur. La différence de temps d'exécution entre les deux entités s'explique par l'insertion de la phase d'apprentissage de la valeur  $k$  au codeur. De plus, nous pouvons vérifier que les chronométrages relevés sont proportionnels aux nombres de valeurs traitées, elles-mêmes en corrélation avec le niveau de la pyramide correspondant.

La série de mesures présentées dans le tableau 9.1 intègre une estimation du paramètre  $k$  réalisée sur l'ensemble des valeurs d'un flux donné. Clairement, cette estimation coûte très cher en termes de temps (de l'ordre de 10 ms par exemple sur le niveau  $l = 0$  de la pyramide). Une solution à ce problème passe par une optimisation du parcours du tableau de données à encoder. Ainsi, nous avons pu remarquer le fait suivant : réaliser l'estimation en ne prenant en compte qu'une valeur sur dix suffit à obtenir un paramètre  $k$  correctement évalué, en comparaison avec celui estimé lors d'une recherche exhaustive. Les temps relevés (voir tableau 9.3) indiquent là aussi une amélioration très nette des résultats.

Chronométrages des fonctions ( $\mu s$ )	
<i>Encodage des erreurs par niveau</i>	
Niveau $l = l_{max} = 4$	45
Niveau $l = 3$	141
Niveau $l = 2$	554
Niveau $l = 1$	2218
Niveau $l = 0$	8832

TAB. 9.3: Relevé des temps d'encodage des erreurs de prédiction par niveau après modification, sur un PC de type Pentium 4, 2.8 GHz, pour la première image de la séquence "foreman\_cif.yuv" ( $352 \times 288$ ).

**Observation finale.** Les modifications (simples) décrites ci-dessus permettent d'obtenir les temps d'exécution des opérations complètes d'encodage et de décodage présentés dans le tableau 9.4.

Chronométrages des fonctions ( $\mu s$ )			
CODAGE	25 334	DECODAGE	9 204

TAB. 9.4: Relevé des temps d'exécution du codec modifié sur un PC de type Pentium 4, 2.8 GHz, pour la première image de la séquence "foreman\_cif.yuv" ( $352 \times 288$ ).

**En conclusion, nous avons à disposition un algorithme (l'Interleaved S+P) non seulement efficace en termes de compression et de scalabilité, mais aussi temps-réel sur PC.** La performance tient également dans le fait que nous réalisons un codage sans perte de l'image.

## 4.2 Cibles multi-PC + multi DSP

Dans ce paragraphe, l'Interleaved S+P a été testé et implanté sur une plate-forme hétérogène multi-composants. Lors de travaux antérieurs, une version simple de la méthode LAR (MICD, codeur entropique Golomb-Rice, post-traitement) a également été intégrée et implantée sur une architecture multi-processeurs (PC et DSP du type C6x) : les chronométrages obtenus sont référencés dans l'article [RBD<sup>+</sup>03]. Avant d'aborder les temps d'exécutions relevés pour

l'Interleaved S+P, nous exposons tout d'abord les différences principales entre un DSP et un PC. Nous présentons par la suite les plate-formes matérielles utilisées dans cette étude.

#### 4.2.1 DSP et PC : comprendre les différences

Les applications de traitement numérique du signal soulèvent essentiellement quatre difficultés [Tis03] :

- un nombre important d'opérations mathématiques sont réalisées et répétées,
- un fonctionnement temps-réel nécessite le respect de contraintes du type entrée/sortie à débit fixe,
- la définition d'un système embarqué (téléphone portable par exemple) impose une basse consommation d'énergie et une taille "raisonnable" des programmes implantés,
- le coût et le temps de développement doivent être minimaux.

Ainsi, convient il de choisir le processeur le plus adapté à l'application visée. Un processeur "généraliste" présente les inconvénients suivants : il consomme trop d'énergie, et est difficilement embarquable. Ces inconvénients disparaissent avec les DSP (Digital Signal Processor), qui apparaissent alors comme les outils les mieux adaptés, en tant que processeurs spécifiques du traitement du signal. Il convient cependant de prendre cette assertion avec quelques précautions : les DSP présentent des limitations en termes de mémoire et de rapidité. En particulier la fréquence de fonctionnement d'un DSP est inférieure à celle d'un Pentium de la génération actuelle. Ceci conduit à la mise en œuvre préalable d'une phase d'optimisation essentielle.

#### 4.2.2 Plate-forme matérielle hétérogène multi-composants

Afin de valider la génération automatique de l'exécutif, les plate-formes construites par le fondeur Sundance<sup>7</sup> ont été choisies. Typiquement, une architecture Sundance est composée d'un PC hôte contenant une ou plusieurs cartes mères, chacune intégrant un ou plusieurs TIM (Texas Instrument Module). Un TIM est un bloc de base à partir duquel le système est élaboré. Il contient un élément de traitement, qui n'est pas nécessairement un DSP : il peut s'agir d'un dispositif d'entrées/sorties, ou encore d'un FPGA<sup>8</sup>. Des mécanismes dédiés aux transferts des données de module à module (bus global, bus SDB<sup>9</sup> 200 MB/s, ComPorts) sont de plus fournis, et implantés sur les TIM au moyen des FPGA.

La plate-forme utilisée dans cette étude s'articule autour d'une carte mère Sundance SMT320. Celle-ci se caractérise par sa modularité, sa flexibilité et sa scalabilité. Elle peut accueillir jusqu'à quatre modules, connectés par bus CP (Com Port) ou SDB. Un FPGA sur chaque module gère six ports de communications (20 MB/s) et deux bus SDB, destinés aux transferts rapides des données (images et informations relatives). Ces forts débits constituent une caractéristique essentielle au regard de la taille importante des données à transférer dans le cadre d'un codage vidéo. La carte mère Sundance se connecte à un PC via un bus PCI.

---

<sup>7</sup><http://sundance.com/>

<sup>8</sup>FPGA : Field Programmable Gate Array

<sup>9</sup>SDB : Sundance Digital Buses

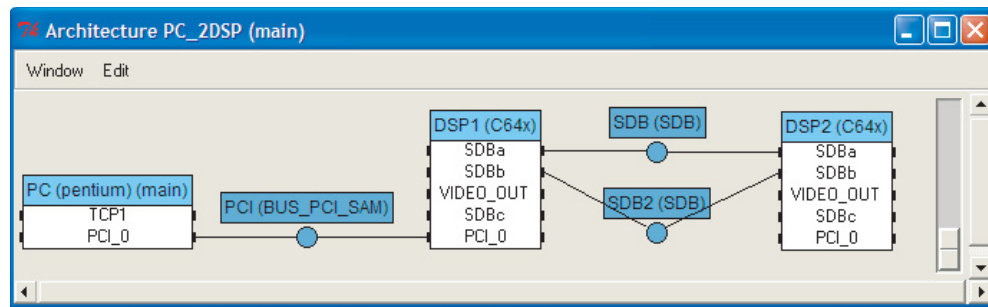


FIG. 9.13: Graphe d'architecture pour deux DSP C64 et un PC.

La modularité de la plate-forme autorise de nombreuses configurations : un exemple de topologie est illustré par la figure 9.13. Le bus PCI de la carte Sundance s'utilise à des fins d'échange de données et autorise une partie des traitements à être exécutée sur PC. Dans cette configuration particulière, la plate-forme multi-DSP peut se concevoir comme un co-processeur du PC.

Le module SMT361 TIM, intégrant un DSP C6416 (fréquence horloge de 400 MHz, 1 MB en mémoire interne) convient parfaitement aux opérations de traitement des images. En effet, la famille C64xx comporte des fonctions spécifiques dédiées à la manipulation des données graphiques.

Le module SMT319 est une "frame grabber", soit une carte d'acquisition numérique et de traitement des vidéos, comportant un DSP C6414 (fréquence horloge de 600 MHz) et deux composants non programmables : un encodeur BT829 (format PAL<sup>10</sup> à YUV<sup>11</sup>), et un décodeur BT864 (format YUV à PAL). Ces deux éléments se connectent au DSP C6414 au moyen de deux FIFO<sup>12</sup>, équivalentes aux bus SDB en termes de débit de données. Le module SMT358 intègre un FPGA (programmable) Virtex XCV600 ainsi que des liens de communications et des blocs IP (de calcul) spécifiques. Cette architecture est représentée sur la figure 9.14. Nous l'utilisons typiquement pour la phase de décodage d'une image. Le DSP C6416 intègre alors toutes les opérations nécessaires à la reconstruction de l'image, alors que le DSP C9414 réalise entièrement la gestion de l'affichage des images décodées.

#### 4.2.3 Prototypage - Résultats d'implantation

Le prototypage du codec Interleaved S+P sur différentes cibles matérielles, et présenté dans ce paragraphe, a été effectué en association avec Mickaël Raulet et David Jacquinet (stagiaire ingénieur INSA).

**Implantation sur un DSP.** Dans un premier temps, nous avons testé l'implantation de notre codeur/décodeur sur un seul DSP du type C6416 (400 MHz), ainsi que sur un DSP du type C6414

<sup>10</sup>PAL : Système normalisé de codage des signaux de la télévision analogique en couleurs caractérisé par l'inversion, une ligne sur deux, de la phase d'une des deux composantes du signal de chrominance, dans le but d'éviter qu'il se produise une erreur de teinte. Il permet d'analyser 25 images à la seconde, comportant chacune 625 lignes de définition, en utilisant la technique du balayage entrelacé. (Source : <http://unskip\penalty\@M://w3.granddictionnaire.com>)

<sup>11</sup>Représentation YUV : Y représente la luminance de la couleur, et U et V, la chrominance de cette couleur dans le rouge et le bleu. Le format utilisé est modulable : par exemple, en configuration dite "4:2:2", 4 valeurs de luminances sont suivies de 2 valeurs U et 2 valeurs V pour coder la couleur d'un seul bloc

<sup>12</sup>FIFO : First In, First Out

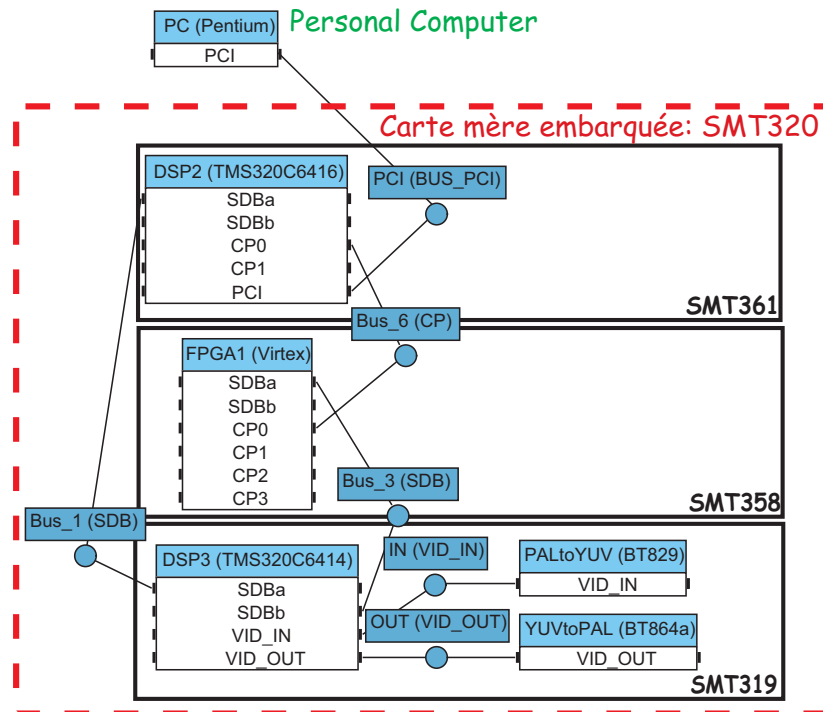


FIG. 9.14: Architecture Sundance utilisée à des fins d'encodage video performant.

(600 MHz, voir description de la "frame grabber"). Le graphe d'architecture correspondant est donné sur la figure 9.15 : le PC ici mis en œuvre se contente d'envoyer les images à coder, puis de recevoir les données issues de la plate-forme afin d'en afficher les images décodées. Du fait de la taille importante de l'espace mémoire nécessaire à l'exécution de l'application (6.1 Moctets pour l'ensemble codeur + décodeur), le compilateur propose de placer par défaut le code en mémoire externe. En effet, seul 1 Moctet de mémoire interne est disponible, contre 32 Moctets en mémoire externe. Une telle implantation entraîne que le temps d'exécution d'un code de cette nature se révèle environ dix fois plus lent en mémoire externe que celui observé en mémoire interne. De ce fait, sur un DSP C6416, la réalisation du codage/décodage de la première image de la séquence "foreman" ( $352 \times 288$  pixels) nécessite 2.4 secondes (1.37 secondes pour le seul codeur).

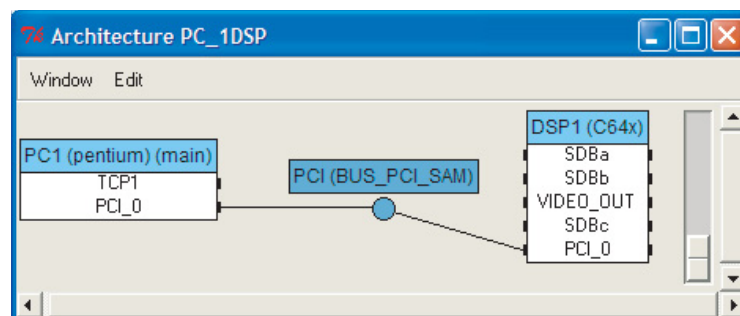


FIG. 9.15: Graphe d'architecture pour un DSP C64 et un PC.

Loins d'être temps-réel, nous avons cherché à calculer le temps d'exécution théorique obtenu si nous avions à disposition un DSP identique au C6416, mais possédant une mémoire interne suffisante. Pour ce faire, il suffit de contraindre l'implantation d'une des fonctions de l'application

(avec une occupation mémoire inférieure à 1 Moctets) en mémoire interne. Le tableau 9.5 recense les différents chronométrages relevés. Il est à noter que les mesures réalisées s'expriment en ticks<sup>13</sup>.

Chronométrages des fonctions (ticks)			
CODAGE		DECODAGE	
Calcul du Quadtree	1 872 288		
Codage de la grille	442 584	Décodage de la grille	2 637 512
Traitement pyramidal	67 672 528	Traitement pyramidal	56 305 928
<i>Encodage des erreurs par niveau</i>		<i>Décodage des erreurs par niveau</i>	
Niveau $l = l_{max} = 4$	161 912	Niveau $l = l_{max} = 4$	43 208
Niveau $l = 3$	498 704	Niveau $l = 3$	159 256
Niveau $l = 2$	1 952 480	Niveau $l = 2$	599 016
Niveau $l = 1$	7 800 400	Niveau $l = 1$	2 322 080
Niveau $l = 0$	$\sim 31\,000\,000^a$	Niveau $l = 0$	$\sim 8\,000\,000^b$
Codage (total estimé)	141 600 896	Décodage (total estimé)	64 791 976

<sup>a</sup>Valeur estimée pour le niveau 0 : la fonction occupant plus de 2 Moctets, il est impossible de l'implanter même seule en mémoire interne.

<sup>b</sup>Voir note a.

TAB. 9.5: Relevé des temps d'exécution des principales fonctions de l'Interleaved S+P sur un DSP de type C64x (utilisé en mémoire interne), pour la première image de la séquence "foreman\_cif.yuv" (352×288).

Sans aucune optimisation de code, nous obtenons au total 141 millions de ticks pour le codeur et 64 millions de ticks pour le décodeur. Si l'on se réfère au DSP C6416 (400 MHz), il s'ensuit que le codeur nécessiterait 278 ms contre 162 ms pour le décodeur. Nous vérifions à nouveau les caractéristiques observées lors de l'implantation sur PC. Ainsi, les différences notables entre codeur et décodeur (traitement pyramidal, encodage et mise en forme des erreurs de prédiction en particulier) trouvent leur explication au paragraphe 4.1.

Clairement, les deux fonctions au cœur du dispositif de codage/décodage (traitement pyramidal au codeur et au décodeur) ne sont pas optimisées pour l'implantation sur DSP, au vu des chronométrages effectués (354 ms au codeur, 175 ms au décodeur sur un DSP C6416 à 400 MHz). Afin de comprendre comment améliorer les performances, il serait intéressant de définir une description plus fine de ces étapes sur SynDEx.

Le calcul du Quadtree, exécuté au codeur uniquement, associé à la phase de compression de l'information codant pour la grille, est réalisé en 5.7 ms sur C6416. Au décodeur, la récupération de la grille nécessite 6.5 ms : une optimisation du code fourni permettrait de diminuer le temps de calcul de cette étape.

Par ailleurs, le problème de la phase de quantification/déquantification (au codeur et au décodeur) demeure (voir §4.1). La suppression de cette opération apporte un gain de temps moins important que ce que nous avons pu constater lors de l'implantation sur PC (de l'ordre de 1 400 000 cycles au codeur, voir tableau 9.6). Les efforts d'optimisation doivent donc être portés sur le code correspondant à la décomposition pyramidale.

<sup>13</sup>Tick : unité de base insécable qui correspond à l'intervalle de temps séparant deux coups d'horloge. Ainsi, si l'on souhaite connaître la mesure exacte d'un chronométrage sur un processeur à une fréquence donnée, il suffit de multiplier le nombre de ticks obtenus par la période de l'horloge prise en compte.



Enfin, les opérations d'encodage par Golomb-Rice sont elles-aussi gourmandes en temps. Cette remarque se vérifie en particulier au codeur, où la phase d'optimisation du paramètre  $k$  ralentit considérablement le processus (voir §4.1). Nous avons donc évalué le temps d'exécution de ces fonctions, dans le cas où la recherche de la valeur de  $k$  se fait de façon non exhaustive (prise en compte d'une valeur sur dix). Le tableau 9.6 résume l'ensemble des temps obtenus après mise en œuvre des différentes modifications précédemment évoquées.

Chronométrages des fonctions (ticks)			
CODAGE		DECODAGE	
Calcul du Quadtree	1 872 288		
Codage de la grille	442 584	Décodage de la grille	2 637 512
Traitement pyramidal	66 291 960	Traitement pyramidal	55 361 864
<i>Encodage des erreurs par niveau</i>		<i>Décodage des erreurs par niveau</i>	
Niveau $l = l_{max} = 4$	59 664	Niveau $l = l_{max} = 4$	43 208
Niveau $l = 3$	191 968	Niveau $l = 3$	159 256
Niveau $l = 2$	734 840	Niveau $l = 2$	599 016
Niveau $l = 1$	2 892 736	Niveau $l = 1$	2 322 080
Niveau $l = 0$	$\sim 12\,000\,000^a$	Niveau $l = 0$	$\sim 8\,000\,000$
Codage (total estimé)	84 486 040	Décodage (total estimé)	63 847 812

<sup>a</sup>Valeur estimée

TAB. 9.6: Relevé des temps d'exécution des principales fonctions de l'Interleaved S+P après modifications sur un DSP de type C64x (utilisé en mémoire interne), pour la première image de la séquence "foreman\_cif.yuv" (352×288).

Pour un DSP à 400 MHz, le codage nécessite 211 ms, le décodage 159 ms. Pour un DSP de la dernière génération à 1GHz, le codage s'effectuerait en 84 ms, le décodage en 63 ms. Ces chronométrages ne constituent qu'une première mise en œuvre de l'Interleaved sur une plate-forme DSP. Il apparaît que des efforts doivent être portés sur l'optimisation du code construit, de telle sorte que notre algorithme puisse non seulement réaliser des performances proches du temps-réel, mais aussi tenir entièrement en mémoire interne du DSP (1 Moctets). **Il est toutefois utile de rappeler que nous obtenons une image codée sans perte**, impliquant de ce fait une application plus complexe que tout autre schéma à encodage avec pertes. Les résultats sont encourageants : **une implantation sur cible embarquée s'envisage désormais**, et, moyennant quelques modifications et optimisations, tout porte à penser que l'application puisse se soumettre aux contraintes temps-réel.

**Evaluation du temps d'exécution sur une topologie à deux DSP.** Nous avons vu l'implantation de l'Interleaved S+P sur une plate-forme ne comprenant qu'un seul DSP. Pour cette étude, nous nous plaçons maintenant dans le cas d'une topologie comprenant deux DSP et un PC, de façon à ce que le codeur (ou le décodeur) Interleaved S+P puisse tirer parti de l'architecture parallèle (figure 9.13). Grâce aux précautions d'écriture du code prises lors de la phase de développement (§3.1), le code correspondant au codec Interleaved S+P se révèle totalement portable et peut être implanté sur cibles multi-DSP. Bien entendu, les résultats d'implantation se limitent à l'utilisation de la mémoire externe disponible sur les DSP, la taille mémoire nécessaire dépassant celle de la mémoire interne.

L'utilisation d'une architecture à deux DSP ne procure pas pour l'instant d'amélioration des temps d'exécution (478 406 680 ticks au codeur, 440 232 168 ticks au décodeur), au regard de ceux recensés lors de l'implantation mono-DSP (422 547 976 ticks au codeur, 417 437 640 ticks au décodeur). L'explication tient essentiellement dans le fait suivant. L'application globale comporte 70 opérations différentes à valuer. A partir de la donnée de la spécification temporelle, SynDEx se révèle capable d'effectuer une meilleure adéquation. Cependant, pour le moment, au niveau de la description SynDEx, les poids relatifs en temps d'exécution de chacune de ces opérations ne sont pas renseignés (même temps d'exécution pour toutes les opérations). Le logiciel fournit par la suite un prototypage sur ces deux DSP loin de l'optimalité : de nouvelles expérimentations sont ainsi envisagées avec l'aide de David Jacquinet.

Afin d'améliorer le dispositif, nous nous devons par ailleurs de donner une description plus fine des étapes de construction et de décomposition pyramidales (sur le modèle de la description du LAR-APP, figure 9.8). Ainsi, nous pourrions exprimer davantage le parallélisme potentiel de l'application : SynDEx serait alors en mesure de réaliser un prototypage exploitant au mieux la topologie donnée.

**Démonstrateurs.** Pour le laboratoire, l'une des finalités des travaux précédemment évoqués réside dans la mise à disposition de démonstrateurs incluant non seulement les étapes de codage/décodage (présentées ci-dessus), mais aussi l'affichage vidéo des images décodées en temps réel. Grâce à la "frame grabber" (figure 9.14), et à la modularité de SynDEx, il s'avère très aisé de porter l'algorithme sur une telle configuration. Pour ce faire, nous avons implanté le décodeur sur le DSP C6414, de telle sorte que l'image reconstruite puisse être visionnée sur un écran vidéo.

## 5 Conclusion

Nous avons présenté dans ce chapitre l'ensemble des étapes utiles à la description sous SynDEx des applications de codage par les méthodes LAR. De premiers résultats de prototypage sur architectures multi-composants ont pu être dégagés en particulier pour l'Interleaved S+P. Le comportement client-serveur de l'application sur architecture cible multi-PC a tout d'abord pu être validé. De plus, **il apparaît que le codeur et le décodeur Interleaved S+P sont tous deux temps-réel pour des images CIF sur PC du type Pentium 4, 2.8 GHz.**

Par ailleurs, nous avons procédé à l'implantation automatique de l'algorithme sur DSP du type C64x. Si l'application n'est pas temps réel, les performances restent toutefois encourageantes, surtout si l'on considère le fait qu'un codage scalable sans perte est réalisé. Une phase d'optimisation du code doit notamment être accomplie afin de porter l'ensemble de l'algorithme en mémoire interne du DSP. Des travaux en cours au laboratoire concernent l'élaboration de méthodes de prototypage rapide implantant automatiquement un mécanisme de mémoire cache : cette innovation permettra un préchargement des données de la mémoire externe vers la mémoire interne. Les contraintes mémoires que nous avons pu relever pourront être de ce fait en partie levées. De plus, il est nécessaire de donner une description SynDEx à granularité plus fine de l'Interleaved S+P, de telle sorte que l'on puisse en extraire au maximum le parallélisme potentiel à exploiter à des fins de prototypage sur architecture cible multi-DSP.



Enfin, il nous a été donné l'occasion de développer une application de démonstration présentée en annexe II, et destinée en particulier à être utilisée dans un cadre de télémedecine. La structure du codeur implanté a été élaborée à partir du code généré automatiquement par SynDEx. La genericité de ce logiciel, exploitée lors de la conception de la plate-forme, justifie une fois de plus son utilisation.

Quatrième partie

# Compression de séquences d'images par la méthode LAR



## Chapitre 10

# Estimation du mouvement et extension de la méthode LAR à la vidéo

L'étude du passage de la méthode LAR à la vidéo aurait dû faire l'objet d'un plus ample développement. Cependant, les résultats obtenus lors de cette thèse, en compression sans perte d'images fixes notamment, ont repoussé les recherches sur l'application du LAR aux séquences d'images. C'est pourquoi nous ne trouverons pas dans ce chapitre de techniques abouties : le travail présenté ici reste prospectif, et permet simplement de soulever les problèmes que nous auront à résoudre.

Une première partie concerne l'état de l'art des méthodes d'estimation de mouvement. Le codage de séquences d'images introduit une donnée supplémentaire exploitable : la redondance temporelle. Ainsi, la modélisation de l'information du mouvement joue un rôle clé dans l'efficacité d'un schéma de compression vidéo. La chaîne de codage intègre donc deux modules additionnels, à savoir : l'estimation de mouvement associée à son encodage (au codeur), et la compensation de mouvement, présente au décodeur (figure 10.1). Ces nouvelles étapes nécessitent la définition au préalable d'un modèle de mouvement (ou encore représentation analytique du mouvement), d'un critère d'estimation, ainsi qu'une stratégie d'optimisation des paramètres de mouvement.

Ce chapitre n'a pas pour prétention de présenter un état-de-l'art complet des techniques d'estimation du mouvement. Il s'inspire d'ailleurs largement des cours dispensés par Nicholas Beser "Image Compression and Packet Video"<sup>1</sup> [Bes03], ainsi que du tutorial "Estimating Motion in Images Sequences" [SK99], réalisé par Christoph Stiller et Janusz Konrad.

Nous avons donc cherché à déterminer les principes fondamentaux qui régissent les algorithmes d'estimation du mouvement, composantes essentielles des standards du type MPEG-4 ou H264 [OBL<sup>+</sup>04]. Si l'exposé balaye un large spectre de méthodes, l'accent sera toutefois mis sur les techniques hiérarchiques basées régions.

En effet, nous avons l'intention de décrire une technique de codage de séquences d'images fondée sur l'exploitation de la partition en régions décrite au chapitre 2. Après avoir vérifié la stabilité temporelle de notre algorithme de segmentation, nous introduisons en section 6 les principes généraux de l'algorithme que nous souhaitons mettre en œuvre.

---

<sup>1</sup><http://apl.jhu.edu/Notes/Beser/525759/index.html>

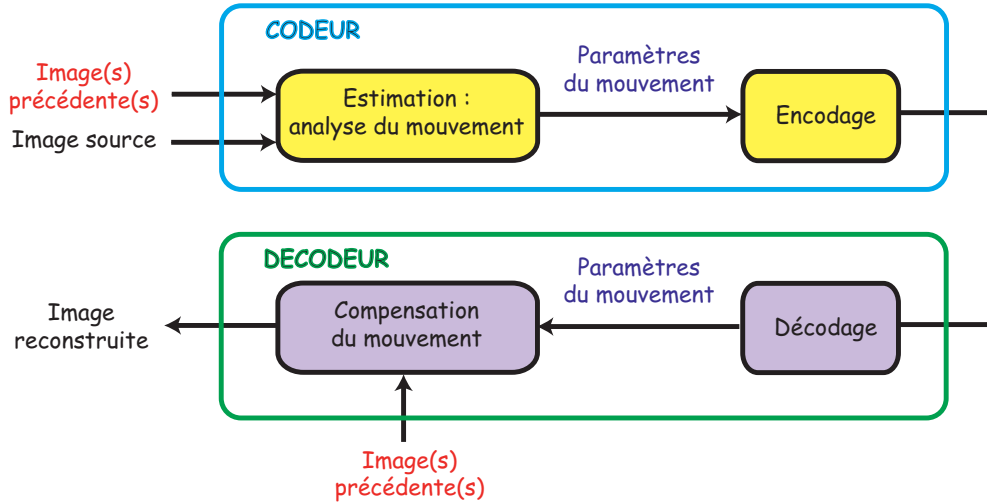


FIG. 10.1: Insertion des blocs d'estimation et de compensation de mouvement dans la chaîne de codage.

## 1 Formation du mouvement - Flot optique

Du point de vue de l'analyse d'image, le mouvement correspond à une variation de l'intensité lumineuse entre deux images d'une même séquence. Cette variation reflète le mouvement relatif de la caméra et des objets présents dans la scène. La phase de modélisation de mouvement implique inévitablement une dépendance vis-à-vis de l'application visée (compression, tracking...), et du contexte de l'étude [MRM03].

Dans cette section, nous nous proposons de mettre en exergue les différences fondamentales qui existent entre le mouvement 2D, et le flot optique.

### 1.1 Formation de l'image et Mouvement 2D

Une séquence vidéo est une représentation plane d'une scène évoluant en 3D. Le mouvement 3D des objets et de la caméra introduit un mouvement 2D dans l'image projetée. Le champ de vecteurs résultant est dépendant des composantes de rotation et de translation induites par les mouvements réels 3D (figure 10.2).

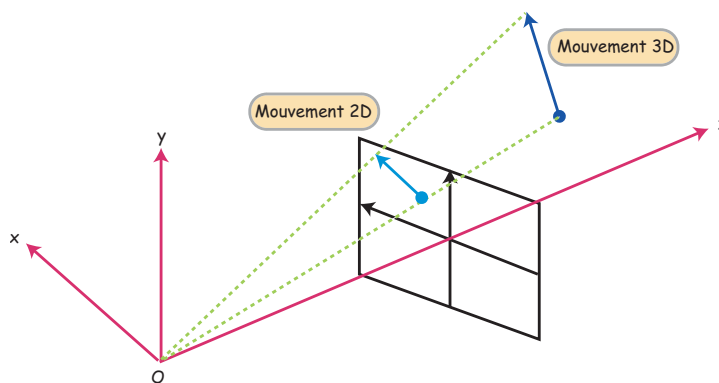


FIG. 10.2: Projection d'un mouvement 3D sur une image plane.

Pour une séquence donnée, il est rare de connaître les paramètres de la caméra à l'origine de la prise de vue. Ceci, ajouté au fait que les trajectoires des objets observés ne sont pas non plus connues, rend impossible une mesure directe des champs de mouvements 2D !

## 1.2 Le flot optique : définition

Comme tout système biologique, l'œil humain est capable de percevoir des mouvements continus infimes. Cette information continue, à l'entrée de tout système de perception visuelle, est appelée **flot optique**. Ce champ de vitesses instantanées affecte à chaque point du champ visuel une "vitesse rétinienne" 2D [BB82]<sup>2</sup>.

Horn et Schunck ont donné la définition fondamentale suivante [HS93, HS81] :

**Définition 10.1 (Flot optique et Mouvement 2D)** *Le flot optique est un champ de vecteurs vitesse de l'image, qui transforme une image donnée en l'image suivante d'une même séquence. [...] Le champ de mouvement est un concept purement géométrique, sans aucune ambiguïté - c'est la projection sur l'image de vecteurs de mouvements 3D.*

## 1.3 Mouvement apparent et mouvement réel

La variation d'intensité lumineuse d'une image 2D n'est pas due au seul mouvement 3D. L'illusion d'optique induite par *l'enseigne de barbier* en est un exemple. La figure 10.3 illustre ce cas d'école : si le mouvement réel du cylindre est bien perpendiculaire à son axe, l'œil perçoit un mouvement du bas vers le haut ! Le lien entre mouvement réel et flot optique (ou encore **mouvement apparent**) n'est donc pas direct.

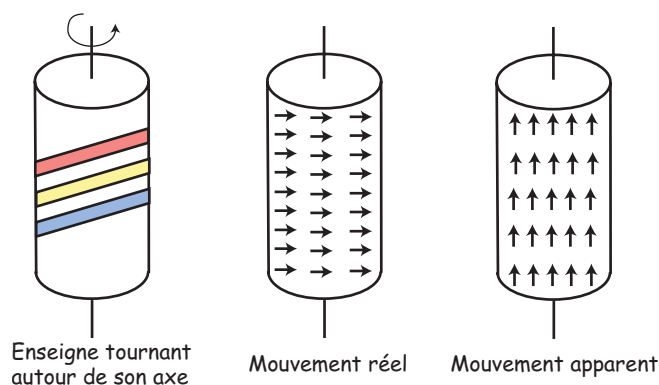


FIG. 10.3: Différence entre mouvement réel et mouvement apparent : cas de *l'enseigne de barbier*.

Par ailleurs, le mouvement apparent dépend aussi des variations de l'éclairage ambiant, ou encore de la texture de surface des objets observés. Prenons l'exemple du cône représenté sur la figure 10.4. L'illustration de gauche montre un cône en rotation, soumis à un éclairage constant : la séquence vidéo résultante est une succession d'images identiques. Sur l'illustration de droite, le cône est cette fois-ci immobile, alors que la source lumineuse éclairant l'objet tourne autour de celui-ci : un "mouvement" est alors détecté.

<sup>2</sup>Le livre référencé [BB82] est disponible en ligne sur le site <http://unskip.penalty@M://homepages.inf.ed.ac.uk/rbf/BOOKS/BANDB/LIB>

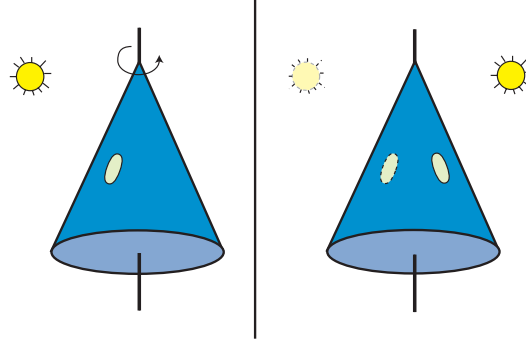


FIG. 10.4: Mouvement apparent induit par une variation d'illumination.

D'une façon générale, une ombre, l'occultation d'un objet présent dans la scène, tout changement brutal d'intensité lumineuse est interprété comme étant issu d'un mouvement par la théorie du flot optique. Le site [http\unskip\penalty\@M://lite.bu.edu/vision/applets/Motion/Motion/Motion.html](http://unskip.penalty@M://lite.bu.edu/vision/applets/Motion/Motion/Motion.html) propose des animations répertoriant les effets visuels remarquables imputables à la différence entre mouvement réel et mouvement apparent.

C'est bien ce mouvement apparent qui doit être évalué à partir de l'information d'intensité lumineuse contenue dans une séquence d'images. Pour ce faire, le flot optique est considéré -sous certaines conditions- comme étant une approximation du champ de mouvements 2D, hypothèse se révélant correcte dans la grande majorité des cas.

#### 1.4 Equation de Contrainte du Flot Optique (ECFO) ou Equation de Contrainte du Mouvement Apparent (ECMA)

L'équation du flot optique modélise la relation existante entre le champ de mouvements 2D et le flot optique proprement dit. Soit  $I(x, y, t)$  un pixel de l'image  $I$ . La vitesse associée à ce pixel s'écrit :

$$v_{x,y} = [v_x, v_y]^T = \begin{bmatrix} dx/dt \\ dy/dt \end{bmatrix}.$$

L'hypothèse de conservation de l'intensité lumineuse du pixel, pendant la durée  $dt$ , est posée, soit

$$I(x + v_x dt, y + v_y dt, t + dt) = I(x, y, t). \quad (10.1)$$

Cette assertion est validée si la scène est éclairée par une source lumineuse constante, et si la surface de l'objet considéré réfléchit la lumière de façon indépendante du mouvement relatif de la scène par rapport à la caméra (surface *lambertienne* [Man03]).

Il est alors possible d'écrire, si l'intensité varie peu en fonction de  $x, y$  et  $t$ , le développement en série de Taylor associé à la partie gauche de l'équation 10.1.

$$I(x, y, t) + \frac{\partial I}{\partial x} v_x dt + \frac{\partial I}{\partial y} v_y dt + \frac{\partial I}{\partial t} dt + O(dt^2) = I(x, y, t),$$

soit encore

$$\nabla I \cdot v_{x,y} + \frac{\partial I}{\partial t} = 0, \quad (10.2)$$

avec  $\nabla I = \left[ \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right]^T$  le gradient de l'image au pixel  $I(x, y, t)$ .

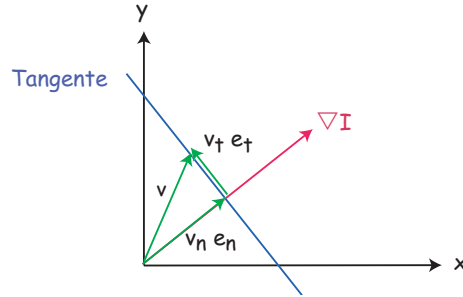


FIG. 10.5: Visualisation géométrique de l'équation du flot optique. Le flot optique d'un pixel donné de l'image peut être n'importe lequel des points situés sur la tangente.

Cependant, les paramètres  $v_x$  et  $v_y$  constituent les deux inconnues de l'équation 10.2. De ce fait, le flot optique ne peut être déterminé par la seule résolution de cette équation. En effet, seul le flot dit "normal" (le long du gradient observé) peut se calculer. Le flot tangent à la direction d'un contour d'intensité égale reste indéterminé (voir figure 10.5). C'est le problème dit d'*ouverture* : pour une zone homogène de l'image, invariante par translation dans le temps, plusieurs mouvements donneront localement la même séquence. Des contraintes supplémentaires globales ou locales, spatiales ou spatio-temporelles, doivent donc être employées afin de récupérer l'ensemble des données du flot optique [Wu04].

Enfin, si le développement de Taylor constitue la technique la plus usitée, il existe d'autres méthodes de décomposition du signal par projection sur une base de fonctions. En particulier, la thèse d'Eric Bruno [Bru01] fait mention de modèles de mouvement basés sur les séries de Fourier ou les ondelettes.

## 2 Estimation du mouvement : méthodologie générale

D'une façon générale, les algorithmes d'estimation de mouvement peuvent appartenir à deux catégories [Bes03] : les méthodes basées primitives (typiquement dans le cadre du "tracking", de la reconstruction 3D...), et les méthodes basées sur l'analyse de l'intensité (codage vidéo). Stiller et Konrad [SK99] ont donc mis en place une méthodologie générale permettant la définition d'un algorithme d'estimation de mouvement. Ainsi, le calcul de la trajectoire des mouvements détectés s'effectue en trois étapes, répertoriées ci-après.

### 2.1 Représentation du mouvement.

Deux modèles sont essentiels à l'estimation du mouvement : le modèle de mouvement, *i.e.* la représentation du mouvement dans la séquence, et le modèle d'observation, permettant d'établir un lien entre les paramètres et les intensités lumineuses de l'image (voir §1.4).

Des modèles de mouvement (représentation, région de l'image) sont donc tout d'abord sélectionnés. Le choix opéré est intrinsèquement lié à l'application visée.



### 2.1.1 Modèle du mouvement.

Modéliser un mouvement est une opération complexe, dans la mesure où le mouvement observé (le mouvement apparent) est une combinaison de projections des mouvements 3D des objets de la scène et des mouvements 3D de la caméra. Toutefois, deux cas simples extrêmement usités ont été construits [Kon00].

Pour une projection orthographique (ou parallèle), une surface 3D quelconque opérant une translation 3D, la vitesse 2D instantanée s'exprime par :

$$v_{x,y} = [v_x, v_y]^T = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \quad (10.3)$$

où les paramètres  $b_1, b_2$  dépendent de la géométrie liée à la caméra et les paramètres de translation 3D. Ce modèle s'avère efficace dans le cadre de la compression de séquences d'images naturelles.

Le modèle *affine* est un modèle de projection associée à un mouvement 3D affine d'une surface plane. La relation suivante décrit les 6 paramètres alors mis en jeu :

$$v_{x,y} = [v_x, v_y]^T = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} + \begin{bmatrix} b_3 & b_4 \\ b_5 & b_6 \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix}. \quad (10.4)$$

Le modèle translationnel précédemment cité est donc un cas particulier du modèle affine. Des modèles plus complexes existent (modèle quadratique, polynômial [Mor98] de degré supérieur à 2). Par exemple, le modèle quadratique s'écrit :

$$v_{x,y} = [v_x, v_y]^T = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} + \begin{bmatrix} b_3 & b_4 \\ b_5 & b_6 \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{bmatrix} b_7 & b_8 & b_9 \\ b_{10} & b_{11} & b_{12} \end{bmatrix} \begin{pmatrix} x^2 \\ xy \\ y^2 \end{pmatrix}. \quad (10.5)$$

Ces modèles n'améliorent cependant pas toujours la précision des mouvements décrits. Il est en effet nécessaire de définir un compromis entre la complexité du modèle et la taille des régions concernées.

Toutefois, la validité du modèle affine (ou même quadratique) n'est pas toujours vérifiée dans le cadre de l'estimation d'un mouvement global : généralement, le mouvement détecté est de nature beaucoup plus complexe. C'est pourquoi il est nécessaire de définir dans l'image des zones pour lesquelles l'hypothèse précédente est correcte. Le partitionnement alors utilisé se base sur l'élaboration d'algorithmes de segmentation, ou encore sur l'estimation hiérarchique des mouvements dominants successifs [Bru01].

### 2.1.2 Support spatial

Il est essentiel de déterminer le support "physique" sur lequel la méthode d'estimation de mouvement doit s'appuyer. La figure 10.6 illustre les différentes approches communément utilisées. Le support spatial du mouvement peut être l'image entière, un pixel de l'image, un bloc rectangulaire de pixels, une région de forme arbitraire.

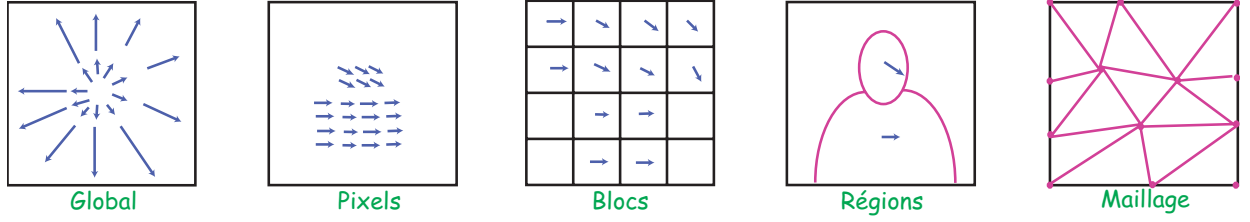


FIG. 10.6: Représentation du mouvement : mouvement global, basé pixels, blocs, régions ou maillage.

## 2.2 Critères d'estimation.

Cette deuxième étape tente de répondre à la question : comment estimer les paramètres du mouvement ? De nombreuses méthodes sont à la disposition des traiteurs d'images : en voici les principales.

**Minimisation de la différence d'images déplacées (DFD) - Appariement.** Soit  $B$  la région de l'image (pixel, bloc, région, l'image entière) au temps  $t$ , que l'on cherche à projeter au temps  $t + 1$ .  $K$  désigne l'espace de recherche, où un appariement est susceptible de se produire ( $K \subsetneq I$ ). On définit la mesure d'appariement liée au déplacement par  $(\delta_x, \delta_y)$  :

$$DFD_{x,y}^t(\delta_x, \delta_y) = \sum_{(b_1, b_2) \in B} (I(x + b_1, y + b_2, t) - I(x + \delta_x + b_1, y + \delta_y + b_2, t + 1))^p, \quad (10.6)$$

avec  $p \in \{1, 2\}$  ( $p = 1$  : MAD<sup>3</sup>,  $p = 2$  : MSE<sup>4</sup>). La solution du flot optique est donnée par la relation suivante :

$$(v_x^t, v_y^t) = \arg \min_{(\delta_x, \delta_y) \in K} DFD_{x,y}^t(\delta_x, \delta_y). \quad (10.7)$$

Notons qu'il existe des méthodes itératives, permettant de minimiser la différence  $DFD_{x,y}^t$  [Odo94]. Le vecteur de paramètres de mouvement est alors réécrit comme étant la somme d'une estimation (initialement égale à 0) et d'un résidu : le problème revient à estimer le résidu. Le processus est répété jusqu'à convergence de l'algorithme (§2.3.4).

**Résolution de l'ECMA (Equation de Contrainte du Mouvement Apparent).** D'une manière générale, on cherche alors à satisfaire l'équation du flot optique et trouver

$$\min \left( C_{x,y}^t = \sum_{(b_1, b_2) \in B} \left| \nabla I \cdot v_{x+b_1, y+b_2}^t + \frac{\partial I}{\partial t} \right|^p \right), \quad (10.8)$$

avec  $p \in \{1, 2\}$  [Man03]. Afin de régulariser le problème mal posé par hypothèse de champ lisse de déplacement, on ajoute une *contrainte de régularité*  $R_{x,y}^t$ . Le problème se réduit donc à la minimisation d'une fonction de coût :

$$\min \left( C_{x,y}^t + \lambda R_{x,y}^t \right), \quad (10.9)$$

avec  $\lambda$  facteur de pondération.

<sup>3</sup>MAD : Mean Absolute Difference

<sup>4</sup>MSE : Mean Square Error

Ce critère convient pour l'estimation de mouvements de faible amplitude. Souvent, il permet la résolution analytique du problème du flot optique. Toutefois, pour des mouvements de grande amplitude, il est possible d'itérer le processus de résolution de l'équation du flot optique, afin de satisfaire le critère DFD.

**Critères stochastiques.** Les méthodes stochastiques sont basées sur des tirages aléatoires pilotant l'exploration des paramètres du mouvement, dans le domaine de recherche  $K$ .

Le critère bayésien, ou Maximum a posteriori (MAP), se propose de maximiser la probabilité

$$\max (P(V = v_{x,y}|I(x, y, t), I(x, y, t + 1))). \quad (10.10)$$

Ainsi, du point de vue de la compression de données, ce critère s'apparente à la recherche du champ de mouvement minimisant la longueur globale du flux binaire pour un encodage sans perte de la séquence d'images [SK99].

D'autres méthodes appartenant à cette famille utilisent les modèles de Markov, ou encore les algorithmes génétiques [Mar98].

**Critère fréquentiel.** Il existe dans la littérature des approches travaillant dans le domaine transformé. Ces méthodes consistent à réaliser une analyse fréquentielle, en particulier par l'emploi de la transformée de Fourier [Ber99]. Plutôt que d'exploiter la propriété de translation/déphasage, les techniques se basent sur la corrélation de phase [Man03]. Si la méthode se révèle robuste et rapide, elle ne peut fournir qu'une information sur le déplacement global constaté sur l'image. Un état-de-l'art de ces techniques, ainsi qu'une solution originale d'estimation de mouvement par l'exploitation de la phase induite sont donnés dans la thèse de Magnus Hemmendorff [Hem01].

## 2.3 Méthodes d'optimisation.

Les stratégies d'optimisation mises en œuvre dans les algorithmes d'estimation/compensation de mouvement visent à proposer un parcours "intelligent" de l'espace de recherche  $K$ . Ainsi, on espère converger rapidement vers un extremum local (minimum ou maximum selon le critère choisi), solution du problème posé. Cette sous-section vise à présenter les méthodes d'optimisation des critères DFD, via en particulier les techniques d'appariement de régions.

De plus, le propos n'étant pas axé sur les approches par maillages, nous avons exclus volontairement l'ensemble des méthodes d'optimisation classiquement appliquées dans ce cadre.

### 2.3.1 Recherche exhaustive

L'idée consiste à parcourir l'ensemble des solutions possibles, puis de sélectionner le meilleur jeu de paramètres. Cette approche est surtout utilisée dans le cas du critère différentiel DFD pour  $p = 1$  (MAD). Si la méthode garantit l'obtention de l'optimal global, le coût de calcul devient rapidement prohibitif, lorsque le nombre de paramètres à estimer simultanément s'accroît. Des algorithmes rapides ont par ailleurs été décrits (voir paragraphe suivant) : la solution retenue est cependant sous-optimale.

### 2.3.2 Matching : Recherche non-exhaustive - Algorithmes rapides

Le *Matching* suit le principe suivant : dans un espace de recherche donné, les valeurs prédites et compensées en mouvement de chacun des candidats possibles sont comparées (matched) avec celles de l'image originale issue de la zone (pixel, bloc, région...) traitée. Le candidat retenu réalise la meilleure estimation au sens du critère considéré.

Une technique très courante consiste en une recherche dite *logarithmique* de candidats à la solution du problème d'appariement de régions [Kon00]. On suppose alors que l'espace de recherche  $K$  est de la forme  $K = 2^n - 1$ . Soit  $K_l = (K + 1)/2^l$ , espace de recherche sous-échantillonné de  $K$ , contenant les points sommets, les points situés au milieu de ces sommets, ainsi que le point central de l'image résolution moitié de l'image originale. Le principe de l'algorithme est le suivant : la première étape consiste à rechercher un candidat sur  $K_1$ . Une fois le meilleur appariement trouvé, une nouvelle zone de recherche  $K_2$  est définie, et la procédure est répétée jusqu'à obtention du "meilleur" candidat (figure 10.7).

D'autres méthodes classiques ont fait leurs preuves : Modified Motion Estimation Algorithm (MMEA), Conjugate Direction Search (CDS), Hexagonal Search, Diamond Search and Three-Step Algorithm (TSA). La figure 10.7 illustre ce dernier.

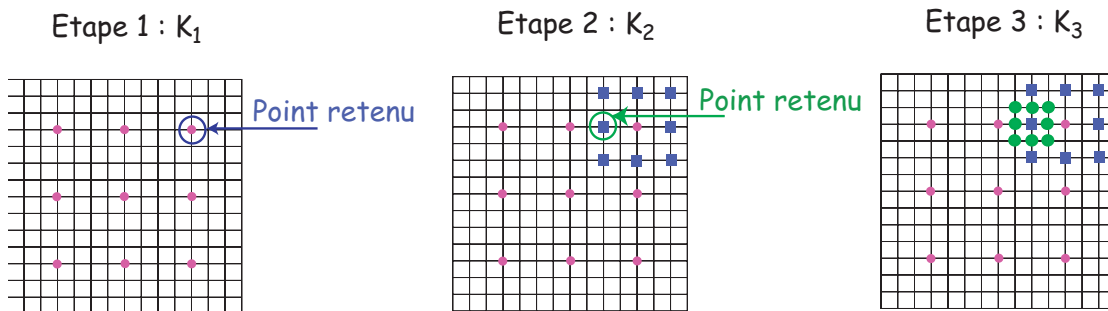


FIG. 10.7: Recherche logarithmique en 3 étapes du bloc candidat à l'appariement (Three Step Search)

### 2.3.3 Stratégie séquentielle

Pour tout point de  $K$ , dans le cadre de critères différentiels, les différences observées sont additionnées, tant que la somme globale reste inférieure à un seuil préalablement fixé. Le "meilleur" déplacement correspond à celui pour lequel le plus grand nombre de point de points a été pris en compte.

### 2.3.4 Descente de gradient

Les techniques d'optimisation par descente de gradient sont tout particulièrement bien adaptées aux critères DFD et celui du flot optique. Le minimum local le plus proche de la solution initiale est alors atteint : à condition d'avoir correctement évalué les conditions initiales et le pas de l'algorithme (voir figure 10.8), il est possible d'obtenir le minimum global de la fonction évaluée.

Si les méthodes dites *pixel-récurrentes* ont été les premières techniques décrites, les algorithmes *region-récurrents* sont de loin les plus efficaces du point de vue analyse des mouvements principaux de l'image (les objets d'une scène sont constitués d'un ensemble de points) [CLL95]. Le calcul du déplacement  $\delta = (\delta_x, \delta_y)$  d'une région, par approche gradient, peut alors être décrit par l'algorithme récursif suivant :

$$\tilde{\delta}^{i+1} = \tilde{\delta}^i - \frac{1}{N} \sum_B \epsilon \cdot DFD_{x,y}^t(\tilde{\delta}_x^i, \tilde{\delta}_y^i) \cdot \nabla I(x - \tilde{\delta}_x^i, y - \tilde{\delta}_y^i, t - 1), \quad (10.11)$$

où

- $B$  désigne la région,
- $\epsilon > 0$  est un facteur de convergence,
- $\tilde{\delta}^i = (\tilde{\delta}_x^i, \tilde{\delta}_y^i)$  représente le déplacement de la région estimé à la  $i$ ème itération,
- $N$  le nombre de pixels de la région  $B$ .

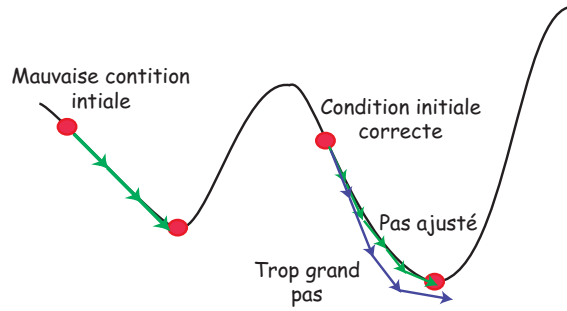


FIG. 10.8: Principe de la descente de gradient.

La descente de gradient au second ordre fait appel à des méthodes classiques du type méthode de Newton (nécessite un gradient d'ordre 2), ou méthode de Newton-Raphelson (approximation du gradient du second ordre par le produit des gradients du premier ordre).

Du fait de l'utilisation de l'approximation par les séries de Taylor, le modèle n'est applicable que sur des mouvements de faible amplitude. Ce désavantage est généralement compensé par une implantation hiérarchique ou multirésolution (voir paragraphe suivant).

### 2.3.5 Méthodes multirésolution

Afin d'éviter de tomber sur un minimum local, et d'accélérer la convergence, les méthodes d'estimation précédemment décrites sont souvent utilisées conjointement avec une technique multirésolution. Le principe de cette technique d'estimation est schématisé sur la figure 10.9.

Les informations issues des images à différentes résolutions spatiales sont combinées, afin d'obtenir à la fois une mesure des vitesses de grande amplitude et une bonne résolution spatiale du flot optique. La technique de raffinement de la mesure du mouvement à partir d'un niveau grossier jusqu'à un niveau fin de résolution est dénommée *coarse to fine strategy* [Bru01].

L'estimation opérée sur le niveau le plus haut de la pyramide ainsi construite ne donne qu'une approximation grossière des plus grands déplacements observables. Le déplacement entre les images est alors compensé par cette estimation, afin que le mouvement résiduel puisse être mesuré à une résolution plus fine. Le procédé est alors réitéré jusqu'à atteindre le niveau le plus bas de la représentation multirésolution.

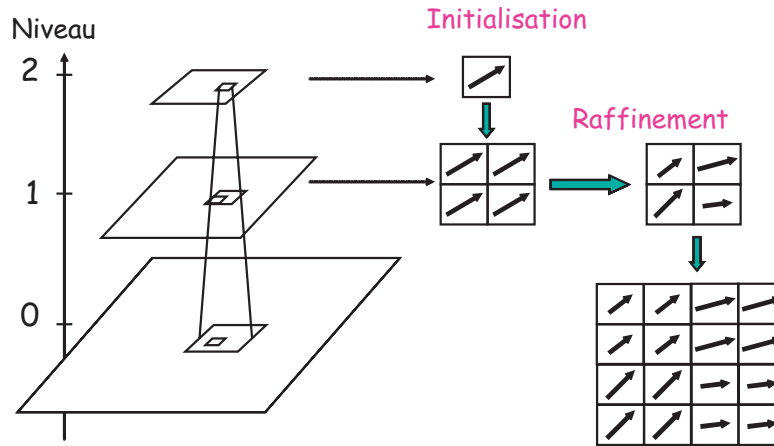


FIG. 10.9: Estimation multirésolution.

Au lieu de contraindre l'exploitation de la pyramide d'estimation du haut vers le bas, il existe par ailleurs des méthodes opérant dans les deux directions. Une boucle de retour permet de réévaluer les erreurs d'estimation de mouvements effectuées sur les basses résolutions, grâce à l'intégration des données rencontrées aux niveaux inférieurs de la pyramide. Si les techniques se révèlent efficaces du point de vue de l'estimation proprement dite, leur implantation se révèle non triviale, et demeure un frein non négligeable à leur développement [SK99].

Il est facile de vérifier que le temps de calcul nécessaire à l'obtention du champ de vecteur de mouvement se trouve nettement diminué, grâce à l'élaboration d'une structure multirésolution. Le principal avantage de l'exploitation d'un schéma hiérarchique est de réduire l'*aliasing temporel* (motifs à structures périodiques d'amplitudes inférieures à celui du déplacement). La multirésolution est naturellement implantée dans les algorithmes d'estimation basés ondelettes; les techniques alors proposées se heurtent à la difficulté de s'affranchir de l'*aliasing spatial*, du à la phase de sous-échantillonnage intrinsèque [YM02].

### 2.3.6 Méthodes hiérarchiques

Si la multirésolution constitue une hiérarchie au sens de la résolution, il est possible de construire un algorithme d'estimation de mouvement basé sur la hiérarchie au sens du contenu de la scène observée. Le plus bas niveau de la hiérarchie est le niveau le plus dense en termes de régions, soit le niveau de granularité la plus forte ou encore de taille de régions la plus faible [Mor98]. Nous reprendrons la définition de la problématique telle qu'elle a été écrite par Franck Morier dans sa thèse précitée :

*"Une hiérarchie de segmentation est créée et évolue au cours du temps. [...] L'empilement de toutes ces régions reconstitue donc l'image. Chaque région peut donc être considérée comme un objet indépendant d'une description plus ou moins raffinée et évoluant au cours du temps."*

Les méthodes opérant sur des maillages peuvent elles aussi s'appuyer sur une représentation hiérarchique. La construction de cette représentation se fait par division/fusion régulière des facettes, ou par ajout/retrait de nœuds et retriangulation [Mar00].

## 2.4 Le sens d'estimation

Ce dernier paragraphe traite du sens de prédiction utilisé à des fins d'estimation/compensation entre deux images successives d'une même séquence. Deux possibilités s'offrent en effet aux traiteurs d'images (figure 10.10). L'estimation de mouvement *avant* (forward) consiste en la recherche, dans l'image  $t$ , d'objets déplacés de l'image à l'instant  $t - 1$ . Dans le cas de l'estimation *arrière* (backward), un objet de l'image courante est recherché dans l'image précédente. Si, pour l'approche *forward*, les paramètres d'estimation doivent être transmis au décodeur afin de réaliser la reconstruction de l'image, l'estimation *backward*, obtenue par analyse de l'image précédemment décodée, ne nécessite pas d'envoi de données supplémentaires [YR00].

La thèse de Chavira-Martinez [CM04] donne un état de l'art de ces techniques par projection arrière ou projection avant. Dans le cadre du suivi d'objet (approches par régions), les premières publications font état d'estimation *forward*. L'idée consiste à projeter à  $t$  les étiquettes des régions définies à l'instant  $t - 1$ . Les régions compensées alors reconstruites à  $t$  sont par la suite réajustées en fonction du contenu réel de l'image. La nouvelle carte de segmentation sert à son tour de base pour celle de l'image suivante. Les problèmes de recouvrement ou de découvrement de régions (deux régions adjacentes n'ont pas le même mouvement global), ou de localisation de frontières sont des limites aux méthodes proposées.

La prédiction *arrière* est désormais la technique privilégiée des approches régions : l'ensemble de l'image est ainsi entièrement recouvert par les régions appariées. Les phénomènes d'occultation par recouvrement ou découvrement sont exclus. En effet, chaque région à l'instant  $t$  trouve un correspondant dans l'image précédente.

Les approches basées maillages sont quant à elles adaptées à la prédiction *avant* : en effet, le contenu de la mémoire peut être exploité à des fins de compensation bien sûr mais aussi pour la définition de partitions dans l'image.

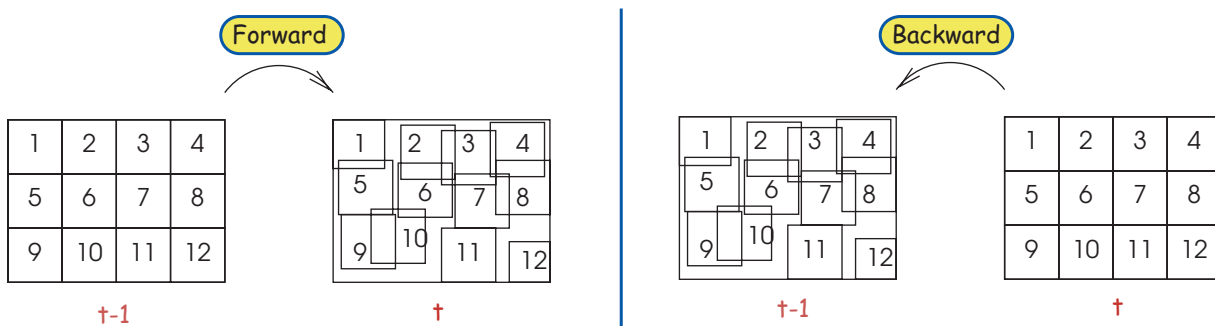


FIG. 10.10: Sens de l'estimation : recherche de vecteurs en arrière ou en avant.

Soulignons toutefois que la norme MPEG attribue la définition inverse aux notions *avant* et *arrière* : lors d'un encodage bidirectionnel, un macro-bloc de l'image  $t$  peut être prédit à partir de l'image  $t - 1$  (*forward*) ou de l'image  $t + 1$  (*backward*).

## 3 Standard MPEG/H.264 : les solutions envisagées

Les standards de compression vidéo (MPEG-4 et H.264) servent incontestablement de référence en matière de compensation/estimation de mouvement. La première partie de cette

section trace les grandes lignes du codage du mouvement dans le schéma H.264<sup>5</sup>. L'algorithme Block Matching (BMA) et ses dérivés (Hierarchical BMA, Deformable BMA), et la prédiction sous-pixelique seront ensuite abordés.

### 3.1 Principe du codage par compensation de mouvement dans H.264

Les informations contenues dans ce paragraphe sont essentiellement tirées de l'article [OBL<sup>+</sup>04]. Dans le cas de prédiction par compensation de mouvement, les macroblocs sont prédits à partir des images précédemment décodées. Pour ce faire, les macroblocs peuvent être divisés en blocs de taille  $16 \times 16$ ,  $16 \times 8$ ,  $8 \times 16$  et  $8 \times 8$ . Dans le cas d'un sous-macrobloc de taille  $8 \times 8$  d'une image dite "P" (prédite), il est possible de le subdiviser à son tour en sous-partitions  $8 \times 4$ ,  $4 \times 8$  ou  $4 \times 4$  (voir figure 10.11). En comparaison, le standard MPEG-4 ne traite que les blocs  $8 \times 8$  ou  $16 \times 16$ .

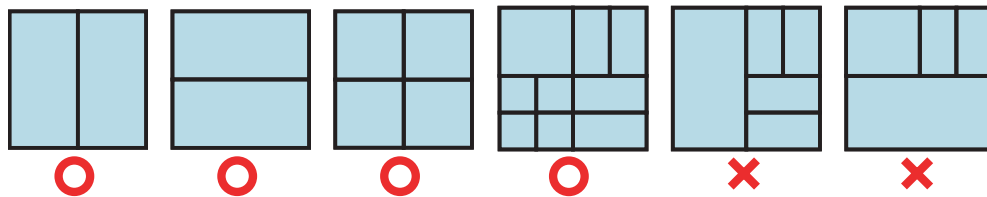


FIG. 10.11: Partitions du macrobloc autorisées (○) ou non (×).

Un vecteur de mouvement est alors estimé et transmis pour chaque bloc. L'estimation peut être réalisée à partir d'une ou plusieurs images précédemment transmises. Une prédiction bidirectionnelle (cas d'une image "B") permet d'utiliser les données issues des images passées et futures (prédiction par combinaison linéaire des signaux *avant* et *arrière*).

Les techniques sous-jacentes, le **Block Matching**, et l'estimation **subpixel**, seront développées dans les paragraphes suivants.

### 3.2 Block Matching et ses dérivées

La technique du Block Matching constitue assurément l'outil privilégié adapté à l'estimation de mouvement. De nombreux travaux ont permis de réaliser des algorithmes efficaces en termes de coût de codage, de robustesse, et de qualité d'images. Ce paragraphe propose un bref aperçu de ces méthodes par appariement de blocs.

#### 3.2.1 Algorithme simple - BMA

Le Block Matching s'appuie sur l'hypothèse suivante : tous les pixels d'un même bloc se déplacent selon un mouvement cohérent. Les paramètres de chaque bloc sont évalués de façon indépendante, par la minimisation de la DFD. Une recherche exhaustive est employée, en particulier si l'on ne cherche qu'un seul vecteur mouvement pour le critère MAD ( $p = 1$  dans l'équation

<sup>5</sup>Le codec H264 a été intégré à la norme MPEG-4 et en constitue la partie 10. La technique est indifféremment nommée H264 ou MPEG-4 AVC (Advanced Video Coding).



de la DFD). Toutefois, des algorithmes rapides (logarithmique simple ou Three-Step Search) ou *subpixel* (voir paragraphe suivant) sont des alternatives efficaces.

Le Block Matching révèle cependant de nombreux défauts. Tout d'abord, les effets de blocs (discontinuités visibles le long des frontières des blocs), dus à la rigidité du modèle de mouvement, sont très désagréables pour l'œil humain : le schéma déformable BMA, ou les méthodes basées région compensent ce défaut. Par ailleurs, le champ de vecteurs engendré est quelque peu "chaotique" : l'estimation réalisée de façon indépendante bloc par bloc en est la cause. Enfin, dans les zones homogènes, la prédiction réalisée est mauvaise (le mouvement reste indéterminé). Néanmoins, du fait de sa simplicité et de sa capacité à optimiser l'erreur de prédiction, c'est le schéma de prédiction compensée le plus largement implanté [Bes03].

### 3.2.2 Le DBMA

Si le schéma le plus simple suppose un déplacement translationnel, le Block Matching dit déformable (DBMA) autorise, quant à lui, des mouvements plus complexes du type affine ou bilinéaire. La prédiction est réalisée grâce à l'exploitation d'un modèle de mouvement basé nœuds. Les nœuds déformables correspondent généralement aux coins des blocs réguliers, définis au niveau de l'image de référence (voir figure 10.12). Les mouvements des autres points des blocs sont simplement les interpolés des mouvements attribués aux nœuds. La translation, le déplacement affine ou bilinéaire constituent des cas particuliers de ce modèle.

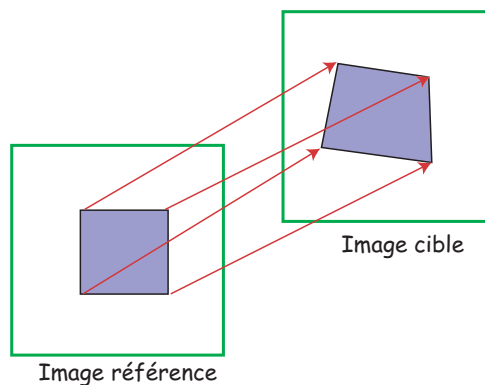


FIG. 10.12: Deformable Block Matching Algorithm : principe.

### 3.2.3 HBMA

Une version multirésolution de l'algorithme du Block Matching a été élaborée (figure 10.13) : à chaque niveau de résolution, on opère un appariement de blocs, en tenant compte des résultats obtenus au niveau supérieur de la décomposition. Les performances de la méthodes ont été évaluées par Naser en particulier [Bes03], et montre un gain substantiel sur la méthode non-hiérarchique.

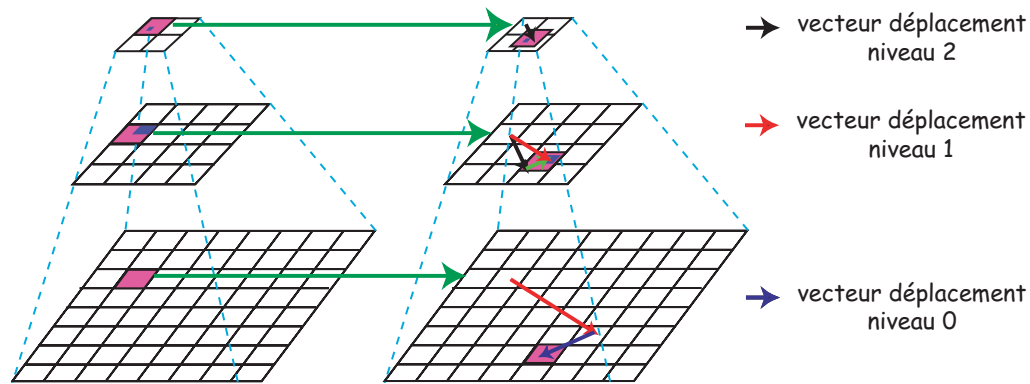


FIG. 10.13: Hierarchical Block Matching Algorithm : principe.

### 3.3 Estimation sous-pixellique

Une des innovations majeures dans le monde de l'estimation de mouvement est sans conteste la prédiction sous-pixellique (*subpixel*). L'amplitude des mouvements réels n'étant en effet pas nécessairement des multiples de pixels, il est souvent plus judicieux de rechercher des mouvements sous-pixelliques, moyennant une étape d'interpolation (typiquement via une interpolation bilinéaire).

S'il est possible de réaliser une estimation à n'importe quelle fraction de pixel, la prédiction au demi-pixel (ou même au quart de pixel pour le standard H264) est la plus utilisée. Le principe est le suivant : la recherche d'appariement s'effectue d'abord en précision entière, pour ensuite être raffinée au demi-pixel.

## 4 Les méthodes multirésolution basées régions : un aperçu

Les méthodes basées blocs, si elles restent simples à implanter (pas de phase de segmentation en particulier), comportent néanmoins de sérieux inconvénients. Entre autres, l'erreur résiduelle suit une distribution à crêtes ("peaky") : elle se concentre en effet au niveau des blocs et des contours de l'image, affectant la qualité visuelle de l'image. Des méthodes de pré- et post traitement améliorent le rendu visuel, mais introduisent un effet de flou typique [YRA99]. C'est pourquoi des solutions basées régions ont rapidement été envisagées.

Rares sont les méthodes d'estimation de mouvement combinant à la fois le mode hiérarchique (au sens des régions issues d'une segmentation donnée) et le mode multirésolution, élaborées à des fins de compression. Nombre de publications font en effet plutôt état de schémas d'estimation/compensation basés régions, dans le cadre plus particulier du suivi temporel. Citons à ce propos les travaux de thèse de Franck Morier [Mor98] et Mehdi Mostafavi [Gor99], sur lesquels nous nous appuierons dans le paragraphe 4.2.

Ainsi, dans cette section, après le relevé de références intéressantes en matières d'algorithmes multirésolution et/ou hiérarchiques, nous essayerons de présenter une méthodologie générale de construction d'une méthode hiérarchique multirésolution de compression vidéo basée région.

## 4.1 Références

Les algorithmes de compression vidéo basés région utilisent la multirésolution dans le cas où les déplacements détectés possèdent une forte amplitude, l'équation de contrainte du mouvement n'étant plus valide. La représentation multirésolution permet de palier à ce problème : au niveau le plus haut de la pyramide construite, les mouvements sont suffisamment peu amples pour que l'hypothèse de petits mouvements, indispensable à l'équation précédemment citée, soit vérifiée. Les travaux d'Odobez et Bouthemy [OB95] constituent la première proposition d'algorithmes en matière de schéma video multirésolution (utilisation d'une pyramide gaussienne).

La représentation multi-niveau peut aussi se définir comme une représentation sémantique de l'image. Par exemple, l'article [ZBK97] propose de dissocier l'arrière plan de l'avant plan d'une scène donnée : la segmentation se résume donc principalement à la description de deux régions différentes. La phase d'estimation/compensation est alors centrée sur l'emploi d'une transformée de Hough, ainsi que de statistiques robustes.

Parmi les travaux relatant de méthodes basées régions associées à une représentation multirésolution et hiérarchique, les approches par maillages semblent les plus nombreuses. La thèse de Gwenaëlle Marquant [Mar00] propose une étude de ces techniques particulières.

La thèse [Tan03] liste quelques articles de référence, avant de présenter une méthode multirésolution basée sur l'exploitation des champs de Markov. Une solution hiérarchique multirésolution a été proposée par Rodriguez et al. [RUGBRSH02], basée sur la génération de pyramides "liées" : deux pyramides, correspondant à deux images successives, sont combinées afin de produire un arbre dont les nœuds sont associés à une région homogène (au sens de la luminance) pour les deux pyramides.

Si les approches par projection arrière sont les plus usitées dans le domaine du traitement basé région de séquences d'images, il existe malgré tout des méthodes travaillant par projection avant. L'article [YRA99] en est une illustration : l'idée principale tient dans le fait que la projection avant d'une image codée à  $t$  évite la transmission de la segmentation de l'image à  $t + 1$  (segmentation dite "*implicite*").

Par ailleurs, l'algorithme élaboré par Morros [MM99] permet d'obtenir une représentation scalable en termes de qualité et de temps ; l'article ne donne toutefois pas de détail sur les "méthodes nombreuses" employées à des fins de codage intra et inter images.

De plus, la définition de la plupart des algorithmes s'appuie sur un *modèle affine* du mouvement. Un précédent travail de Yoon l'avait toutefois conduit à envisager un encodage *forward*, basé sur le modèle translationnel [YRA97].

Enfin, rappelons que chaque algorithme basé région possède une phase essentielle de segmentation. Cette segmentation peut être strictement spatiale (image par image), strictement temporelle ou encore spatio-temporelle. Le lecteur pourra se référer aux articles de Salembier [SM99], Zhang [ZL01], Wanderley [WD02] et Cremers [CS04], établissant un état-de-l'art des méthodes efficaces de segmentation. Notons que la plupart des méthodes introduites exploitent l'information contours.

## 4.2 Méthodologie générale

Afin de préparer l'élaboration de notre propre méthode de codage vidéo, il nous semble important de déterminer une méthodologie de conception d'algorithme scalable en termes de résolution

et de hiérarchies sur les régions. Cette section s'attache donc à présenter les différentes phases de réflexion nécessaires.

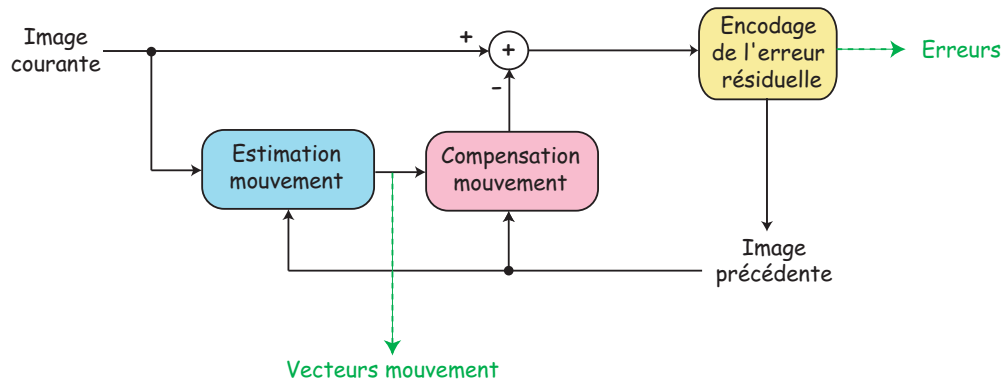


FIG. 10.14: Schéma-blocs d'un codeur simple.

Dans cette optique, partant du schéma de base présenté en figure 10.14, nous étudions dans un premier temps les solutions possibles de parcours de la représentation pyramidale. Le paragraphe 4.2.2 détaille par la suite les étapes principales de construction d'un flux vidéo scalable (associant segmentation multiéchelle, multirésolution...).

#### 4.2.1 Hiérarchie et Multirésolution : stratégie de parcours de la représentation pyramidale

Dans le but d'allier les notions de hiérarchie et de multirésolution, il est indispensable de se fixer au préalable une stratégie de parcours de la représentation pyramidale ainsi construite. La figure 10.15 propose une illustration graphique des différentes solutions possibles.

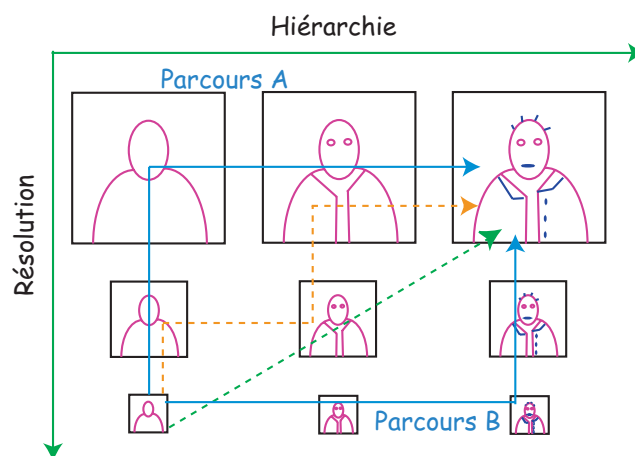


FIG. 10.15: Combinaison des approches hiérarchiques et multirésolution.

Le parcours dénommé "Parcours A" sur le schéma se décompose en deux étapes. Tout d'abord, pour un niveau de hiérarchie donnée, la descente de la pyramide multirésolution permet d'obtenir un encodage scalable en termes de résolution. La seconde étape consiste en l'encodage progressif hiérarchique des régions de l'image. De ce fait, la condition nécessaire à l'estimation de mouvement, à savoir une faible amplitude des déplacements constatés, est garantie à tous les niveaux de notre

décomposition. Le parcours B fonctionne symétriquement au parcours A : la scalabilité en terme de régions est d'abord abordée avant d'exploiter la multirésolution.

Les chemins dessinés en pointillés peuvent être, quant à eux, qualifiés de "hybride" : ils définissent une autre façon de combiner hiérarchie et multirésolution. Par exemple, le parcours vert propose la solution suivante : à chaque niveau de résolution  $R^l$  donné, la représentation de niveau hiérarchique équivalent  $H^l$  est choisie pour l'encodage.

**Ainsi, la méthode idéale serait celle qui se révélerait capable de suivre n'importe quel chemin associant hiérarchie et multirésolution.**

#### 4.2.2 Les étapes de construction du flux vidéo scalable

La figure 10.16 permet de visualiser les principales composantes d'un algorithme complet de codage vidéo scalable. Ce paragraphe a pour but de présenter ces briques élémentaires de la construction de toute méthode d'encodage vidéo par régions.

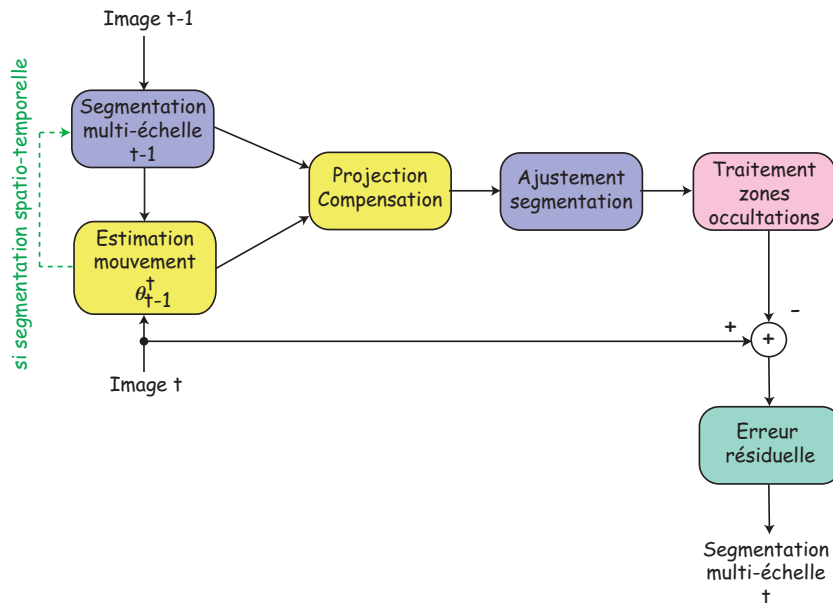


FIG. 10.16: Segmentation itérative - suivi temporel.

**Segmentation multiéchelle.** La segmentation utilisant le mode Inter-image (produisant des images I, dénomination officielle des standards MPEG) se définit comme suit : chaque image d'une séquence donnée est récursivement segmentée moyennant l'utilisation des données de l'image précédente [STMG95]. Si de nombreuses méthodes existent dans le domaine de la segmentation hiérarchique, elle ne sont cependant que rarement associées à une multirésolution. La multirésolution n'est en général pas utilisée en temps que moyen de visualisation scalable du flux codé, mais plutôt comme régulateur de débit [YRA99], ou méthode de convergence rapide [PBK94].

**Estimation de mouvement multirésolution.** Le mouvement observé entre deux images successives pour un objet vidéo donné se calcule par le déplacement moyen appliqué au centroïde<sup>6</sup> de l'objet [CCM<sup>+</sup>97].

L'estimation de mouvement multirésolution peut être couplée à l'étape précédente, dans le cadre d'une segmentation spatio-temporelle (prise en compte de l'information de déplacement afin de construire une partition valide aussi au sens du mouvement). Un algorithme simple est celui proposé par Odobez et al [STMG95] : le MRLS (Multiresolution Least Mean Square Algorithm) et ses dérivées robustes (RMR : Robust Multiresolution Algorithm, basé sur la méthode IRLS : Iteratively Reweighted Least Squares). A nouveau, le lecteur pourra se reporter aux références citées au paragraphe précédent, définissant des algorithmes efficaces de segmentation spatio-temporelle.

**Projection.** La phase de projection traduit l'évolution au cours du temps des régions issues d'une segmentation donnée, et exprime ainsi le lien temporel qui existe entre les régions. L'encodage de la partition courante est réalisé via un mécanisme de prédiction/compensation issu de la partition précédente. Aucune nouvelle région n'est alors introduite au niveau de l'image courante.

**Ajustement de la segmentation.** Connaissant la carte de segmentation prédite par l'étape précédente, il est nécessaire de procéder à un ajustement spatial des régions. Peu d'articles mentionnent en détail cette phase du procédé de codage.

Les travaux successifs de Wu et Morier [Mor98] mettent en œuvre un algorithme exploitant le modèle des "snake polygonaux" (basé contours). Mostafavi propose une amélioration de la procédure de suivi temporel proposée par Wu et Morier : par un ajustement séquentiel [Gor99], les régions sont traitées dans l'ordre décroissant de leur superficie. L'objectif est double : identifier les pixels non étiquetés par prédiction et susceptibles d'appartenir à une région donnée, ou encore affiner les régions produites (pertes de points). L'enchaînement séquentiel d'une phase d'ajustement temporel (optimisation des paramètres du mouvement), et d'une phase d'ajustement spatial, améliore sensiblement la qualité d'estimation du mouvement.

La thèse de Victor Nzomigni s'appuie sur une relaxation mono-échelle déterministe ICM (Iterated Conditional Modes) [Nzo95], qui peut s'apparenter à une descente de gradient. Enfin, citons l'approche par régularisation statistique basée sur les champs de Markov aléatoires multi-échelles établie par Odobez *et. al.* [OB98].

**Traitement des zones d'occultation.** Les zones d'occultation, et en particulier les régions découvertes sont ici examinées. Classiquement, celles-ci subissent une phase de segmentation intra-image. A ce procédé, nous pouvons ajouter celui de fusion éventuelle au sens du mouvement des régions prédites. Dans les méthodes par compensation avant, il est courant d'associer une prédiction arrière complémentaire afin de traiter efficacement ces zones d'occultation [YRA97, YRA99].

---

<sup>6</sup>Centroïde : Point fictif situé à l'intérieur d'un objet et dont les coordonnées correspondent au centre de cet objet

**Encodage de l'erreur résiduelle.** Une image résiduelle est observée, résultat de la différence entre l'image originale et l'image reconstruite après les phases d'estimation/compensation.

Dans ce cadre, une solution avantageuse en termes de coût de codage s'appuie sur le fait que cette erreur se situe plutôt au niveau des frontières des régions translatées, ainsi qu'au niveau des zones de découvrément [YRA99]. Cette propriété permet d'allouer un débit plus important sur les zones à forte concentration d'erreurs.

#### 4.2.3 Un exemple d'algorithme complet : le VideoQ

Nous avons choisi de montrer un algorithme complet de codage vidéo hiérarchique par région : le VideoQ [ZC97, CCM<sup>+</sup>97]. Nous en explicitons ici les grandes étapes successives, représentées sur la figure 10.17.

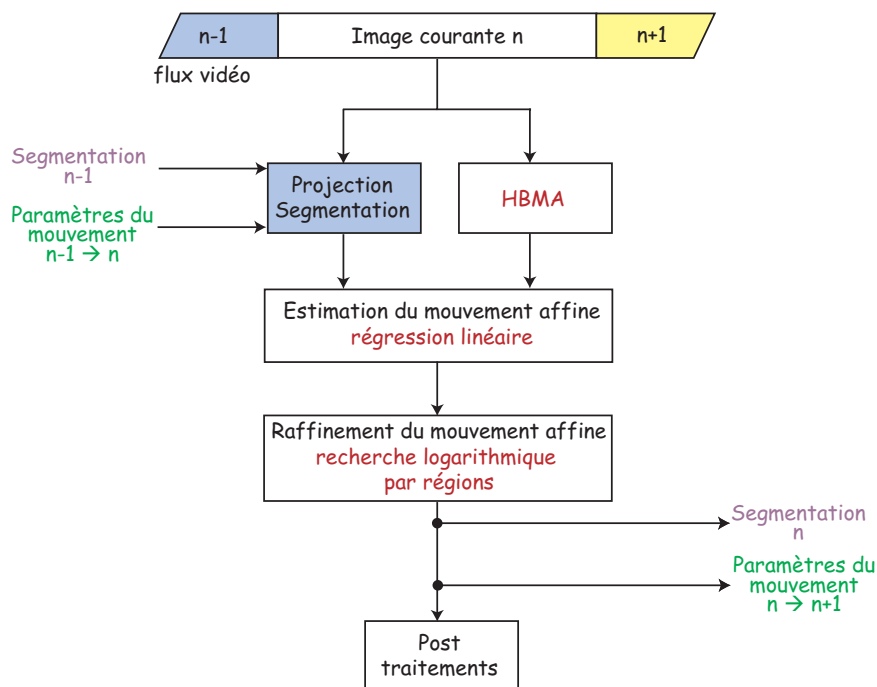


FIG. 10.17: Schéma de réalisation du codeur VideoQ.

Le flot optique est tiré de l'observation du mouvement engendré entre les images  $n$  et  $n + 1$  ; le Hierarchical Block Matching Algorithm (§3.2.3) est utilisé à cet effet. Le module de projection/segmentation réalise une partition à  $n$  prédite à partir de l'image  $n - 1$ . A partir des deux informations précédentes (flot optique et régions couleur), les paramètres de mouvement affine sont estimés via un algorithme de régression linéaire. Ces paramètres sont à nouveau raffinés : une étape d'appariement par régions utilise la recherche logarithmique dans l'espace des six dimensions déterminées par un mouvement affine. Ainsi, les régions, cohérentes du point de vue couleur, associées à un mouvement affine, sont élaborées pour l'image  $n$  du flux vidéo.

## 5 État de l'art : conclusion

Dans ce chapitre, nous avons détaillé la méthodologie communément admise de construction d'un algorithme de codage vidéo par estimation/compensation de mouvement. Si notre état-de-l'art n'est pas exhaustif, il expose les principales publications ayant trait au codage scalable par régions des séquences d'images. En effet, pour nous, l'enjeu actuel de l'élaboration des schémas de codage concerne la mise en œuvre d'outils intégrant la plus grande variété de fonctions scalables possibles. Les recherches réalisées en particulier autour des normes MPEG et DICOM (traitement des images et vidéos médicales) visent à proposer des méthodes combinant approche multirésolution et codage par régions d'intérêt.

En guise d'illustration, l'algorithme proposé par Di Zhong *et al.* (§4.2.3) donne une solution intéressante d'encodage par régions, alliant prédiction hiérarchique au niveau blocs et raffinement de l'information de mouvement au niveau régions.

## 6 Le LAR et la vidéo

Comme nous l'avons indiqué dans la partie introductive de ce chapitre, l'extension de la méthode LAR à la vidéo reste encore à l'état prospectif. C'est pourquoi cette section ne cherche pas à donner de résultats définitifs, mais plutôt à proposer des pistes de réflexion à partir desquelles nous espérons travailler.

Après avoir exposé les objectifs de l'étude, et en particulier l'intérêt de la segmentation associée à un codage de séquences d'images, nous nous intéresserons au comportement de notre décomposition en régions et son évolution temporelle. Enfin, nous évoquerons les principes sur lesquels nous souhaitons appuyer notre schéma de compression.

### 6.1 Objectifs de l'extension du LAR à la vidéo

Nous avons vu dans la partie bibliographique de ce chapitre les éléments essentiels sur lesquels s'appuient l'ensemble des codeurs vidéo, à savoir : une phase d'estimation et de compensation de mouvement, et une phase de codage des erreurs. De l'efficacité de ces étapes dépend étroitement la taille des données à transmettre. Les approches standard utilisent un partitionnement de l'image en blocs de taille fixe (MPEG-4) ou en blocs de taille variable (MPEG-4 AVC) afin de diminuer à la fois le coût de codage des vecteurs de mouvement obtenus et celui des erreurs. Les efforts conjoints d'une grande partie de la communauté des traiteurs d'images spécialisés en compression ont permis d'élaborer des algorithmes extrêmement efficaces, depuis les bas jusqu'aux hauts débits.

Le propos n'est donc pas de réaliser un codage optimal au sens des objectifs atteints par H264, mais plutôt de proposer une méthode intégrant des fonctionnalités avancées encore absentes de ce standard, du type : scalabilité complète (en termes de résolution, de qualité d'image reconstruite, de régions), codage des objets pour une compression à très bas débits, ou encore compression sans perte.

Pour ce faire, nous escomptons réutiliser la technique de segmentation décrite au chapitre 2. Comme nous l'avons mentionné, le principal désavantage des méthodes de compression fondées sur l'exploitation d'une segmentation réside dans le fait que cette dernière repose typiquement sur la description des contours présents dans l'image. Le coût d'encodage de cette information demeure



une limitation importante à l'émergence de ce type de technique. Notre approche s'affranchit de cet inconvénient : la carte de segmentation, auto-extractible, se révèle disponible au décodeur à coût nul, ou encore à coût minimal si elle est couplée à un contrôle par la chrominance. De plus, l'adéquation entre le partitionnement en régions et la forme des objets présents dans la scène apparaît particulièrement forte. Toutes ces considérations nous laissent penser que notre algorithme peut s'affirmer un bon candidat à l'élaboration d'un codage vidéo par régions efficace.

Par ailleurs, les méthodes scalables du LAR, décrites aux chapitres 5, 6 et 7, apportent une valeur ajoutée non négligeable à notre technique. Ainsi, est il possible d'**envisager la définition d'un codage vidéo combinant hiérarchie et multirésolution**.

## 6.2 Etude de la cohérence temporelle de la segmentation

Avant de tenter de réaliser un système d'encodage vidéo par description en régions, il est de bon aloi d'en vérifier la robustesse temporelle. Observons donc le comportement de notre segmentation sur une séquence d'images. Cette étude a été précédée de la réalisation du codec (codeur + décodeur) complet, ainsi que de sa description sous forme de graphe flot de données intégrée dans SynDEx.

Dans les figures 10.18 et 10.19, nous montrons respectivement la représentation en régions et le résultat de l'encodage complet des douze premières images de la séquence "Foreman\_cif.yuv". La luminance est codée par blocs à travers un système prédictif simple (prédicteur médian adapté à notre représentation non-uniforme, voir chapitre 1). Les composantes de chrominance sont encodées par régions, après mise en œuvre de la phase de segmentation : 3 niveaux de fusion successifs avec un seuil  $Th_{Cost}$  prenant consécutivement les valeurs  $\{15, 30, 50\}$ , et un coefficient de contrôle de la chrominance égal à 0.5. Il est à noter qu'aucun post-traitement n'est appliqué à l'image reconstruite au décodeur. Enfin, le coût indiqué correspond au coût réel après encodage par les codes de Golomb-Rice (voir chapitre 9, §4.1) des différents flux codant l'image concernée.

Visuellement, le constat est immédiat : le codage par régions des composantes chromatiques reste cohérent d'une image à l'autre, et ce tout au long de la séquence. Cette cohérence se traduit également, lors de l'observation d'une séquence d'images similaires, par un nombre de régions et un coût de codage stables. De plus, les bonnes propriétés de la segmentation (adéquation entre les formes extraites et le contenu de l'image, coût minimal en particulier) restent valables pour le cas de la vidéo. Ces observations ne se limitent pas à la seule séquence présentée dans la figure 10.19 : les mêmes remarques s'appliquent sur toutes les séquences de test MPEG-4 habituelles que nous avons pu tester.

Ainsi, semble t'il raisonnable de penser que l'algorithme de segmentation couplé au LAR doit pouvoir s'adapter efficacement à des fins de codage vidéo. De plus, la hiérarchie naturelle de la méthode lui confère un atout supplémentaire. Utilisée en association avec un schéma pyramidal, la scalabilité s'exprimerait à la fois en termes de résolution et de qualité, mais aussi en termes de régions. De ce fait, l'ensemble des parcours de la décomposition pyramidale évoqués sur la figure 10.15 sont envisageables.



a. 81 régions (U : 0.004 bpp, V : 0.006 bpp, Contrôle : 0.059 bpp)



b. 88 régions (U : 0.004 bpp, V : 0.006 bpp, Contrôle : 0.059 bpp)



c. 83 régions (U : 0.006 bpp, V : 0.006 bpp, Contrôle : 0.059 bpp)



d. 77 régions (U : 0.005 bpp, V : 0.006 bpp, Contrôle : 0.059 bpp)



e. 85 régions (U : 0.004 bpp, V : 0.006 bpp, Contrôle : 0.060 bpp)



f. 86 régions (U : 0.006 bpp, V : 0.006 bpp, Contrôle : 0.059 bpp)



g. 78 régions (U : 0.004 bpp, V : 0.006 bpp, Contrôle : 0.061 bpp)



h. 89 régions (U : 0.004 bpp, V : 0.006 bpp, Contrôle : 0.060 bpp)



i. 93 régions (U : 0.004 bpp, V : 0.007 bpp, Contrôle : 0.062 bpp)



j. 86 régions (U : 0.005 bpp, V : 0.006 bpp, Contrôle : 0.062 bpp)

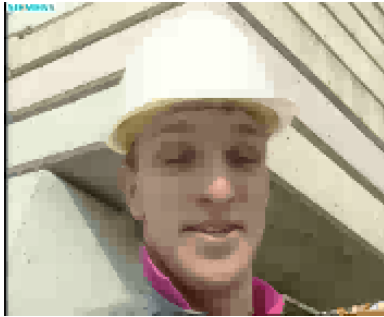


k. 100 régions (U : 0.005 bpp, V : 0.007 bpp, Contrôle : 0.063 bpp)

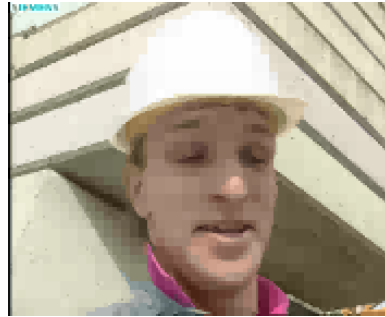


l. 91 régions (U : 0.005 bpp, V : 0.006 bpp, Contrôle : 0.063 bpp)

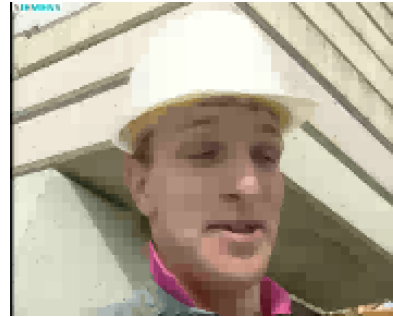
FIG. 10.18: Etude de la segmentation par contrôle de la chrominance sur les 12 premières images de la séquence "Foreman\_cif.yuv" : observation de la représentation en régions.



a. 81 régions, 0.277 bpp  
(Grille : 0.045 bpp, Y : 0.163 bpp,  
U : 0.004 bpp, V : 0.006 bpp,  
Contrôle : 0.059 bpp)



b. 88 régions, 0.282 bpp  
(Grille : 0.045 bpp, Y : 0.166 bpp,  
U : 0.004 bpp, V : 0.006 bpp,  
Contrôle : 0.059 bpp)



c. 83 régions, 0.281 bpp  
(Grille : 0.045 bpp, Y : 0.165 bpp,  
U : 0.006 bpp, V : 0.006 bpp,  
Contrôle : 0.059 bpp)



d. 77 régions, 0.278 bpp  
(Grille : 0.044 bpp, Y : 0.163 bpp,  
U : 0.005 bpp, V : 0.006 bpp,  
Contrôle : 0.059 bpp)



e. 85 régions, 0.281 bpp  
(Grille : 0.045 bpp, Y : 0.165 bpp,  
U : 0.004 bpp, V : 0.006 bpp,  
Contrôle : 0.060 bpp)



f. 86 régions, 0.281 bpp  
(Grille : 0.045 bpp, Y : 0.164 bpp,  
U : 0.006 bpp, V : 0.006 bpp,  
Contrôle : 0.059 bpp)



g. 78 régions, 0.281 bpp  
(Grille : 0.045 bpp, Y : 0.165 bpp,  
U : 0.004 bpp, V : 0.006 bpp,  
Contrôle : 0.061 bpp)



h. 89 régions, 0.283 bpp  
(Grille : 0.046 bpp, Y : 0.167 bpp,  
U : 0.004 bpp, V : 0.006 bpp,  
Contrôle : 0.060 bpp)



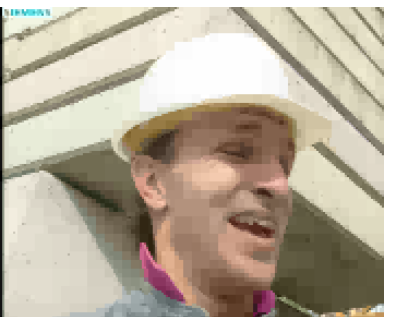
i. 93 régions, 0.289 bpp  
(Grille : 0.046 bpp, Y : 0.170 bpp,  
U : 0.004 bpp, V : 0.007 bpp,  
Contrôle : 0.062 bpp)



j. 86 régions, 0.290 bpp (Grille :  
0.046 bpp, Y : 0.170 bpp, U : 0.005  
bpp, V : 0.006 bpp, Contrôle : 0.062  
bpp)



k. 100 régions, 0.294 bpp  
(Grille : 0.047 bpp, Y : 0.171 bpp,  
U : 0.005 bpp, V : 0.007 bpp,  
Contrôle : 0.063 bpp)



l. 91 régions, 0.298 bpp (Grille :  
0.047 bpp, Y : 0.176 bpp, U : 0.005  
bpp, V : 0.006 bpp, Contrôle : 0.063  
bpp)

FIG. 10.19: Etude de la segmentation par contrôle de la chrominance sur les 12 premières images de la séquence "Foreman\_cif.yuv".

### 6.3 Codage vidéo par le LAR : les méthodes envisagées

Le paragraphe précédent a permis de mettre en lumière les propriétés intéressantes et exploitables pour du codage à très bas débits et avec pertes des séquences d'images. Nous proposons dans ce qui suit un ensemble de techniques plus ou moins complexes fondées sur l'approche conjointe du LAR et de la représentation en régions. Les principes généraux sont essentiellement donnés : le choix de l'estimateur de mouvement n'est pas encore totalement fixé. Néanmoins, dans le sens où nous n'avons pas comme objectif la réalisation de la meilleure estimation possible, nous projetons de mettre en œuvre des techniques (relativement) simples du type HBMA (§3.2.3). Pour des raisons de compatibilité avec MPEG-4, les termes estimation *backward* et *forward* se rapportent à ceux utilisés dans ce standard.

La notation  $I_n$  correspond à la  $n^{\text{ième}}$  image de la séquence, et déjà encodée. Le problème ici posé est le suivant : comment concevoir le codage de l'image  $I_{n+1}$  par prédiction (image "P") ?

La réalisation des méthodes suivantes apparaît conditionnelle à l'obtention de la représentation non uniforme du LAR. Le partitionnement Quadtree se révèle très sensible à toutes les opérations de translation. Il s'avère donc judicieux de préalablement transmettre l'information de grille adaptée à l'image  $I_{n+1}$  : dans le cas contraire, les distortions dues à une déformation de la partition se révéleraient très dommageables.

#### 6.3.1 Compensation rudimentaire et codage des chrominances par région

Une première ébauche de l'extension du LAR à la vidéo consisterait en l'encodage simple de la luminance par blocs après une phase d'estimation du mouvement appliquée sur chacun de ces blocs (la taille minimale des blocs traités étant égale à  $4 \times 4$ ). Cette prédiction de la luminance par blocs, classiquement opérée en mode *forward*, permet le calcul immédiat au décodeur de la carte de segmentation de l'image  $I_{n+1}$ . Ainsi, l'encodage des composantes de chrominances s'effectue au niveau régions.

Cette solution a l'avantage d'être simple à implanter : il existe en effet de nombreuses fournissant les briques de base nécessaires au développement de l'algorithme d'estimation/compensation.

#### 6.3.2 Compensation du mouvement par région, codage de la luminance par bloc

Afin de tirer profit de la segmentation disponible pour l'image  $I_n$ , il est possible de mettre en place une étape de compensation *backward* du mouvement par régions. Ainsi, la prédiction des valeurs de luminance des blocs de  $I_{n+1}$  s'obtient par la projection des valeurs des blocs de  $I_n$ , conformément au déplacement de la région auxquels ils appartiennent. Cette technique devrait avoir l'avantage d'une plus grande insensibilité aux erreurs de prédiction par régions.

Les cas de recouvrements (plusieurs blocs compensés pouvant servir à l'estimation de la valeur du bloc cible), se traitent alors simplement par le calcul d'une moyenne pondérée des valeurs en fonction du taux de recouvrement d'un bloc. Il est à noter que le recouvrement même partiel d'un bloc peut permettre l'attribution d'une valeur prédite. Pour les blocs non estimés, ceux-ci sont parfaitement identifiables, et un processus de codage classique en mode intra (MICD simple ou approche pyramidale) saura les coder efficacement.

Par ailleurs, la quantification adaptée à la taille des blocs devrait ici également conduire à de bons résultats, les plus fortes erreurs de prédiction se retrouvant sur les contours (petits blocs).

Soulignons que la quantification des erreurs de prédiction est liée au partitionnement de l'image  $I_{n+1}$ .

Enfin, de même que pour la technique exposée au paragraphe précédent, l'information de chrominance est encodée par région après obtention du partitionnement à  $I_{n+1}$ .

De sorte à obtenir un résultat encore plus cohérent, nous envisageons d'introduire la notion de mouvement dans l'exécution de l'algorithme de segmentation (**segmentation spatio-temporelle**). Ainsi, partant d'un niveau donné de la hiérarchie de la segmentation, certaines régions peuvent être fusionnées au sens du mouvement, d'autres au contraire nécessitent une redécomposition (pas de vecteurs de mouvement uniformes). L'exploitation de cette projection peut alors servir soit directement pour la prédiction des blocs (comme ici), soit pour la prédiction de la nouvelle carte de segmentation (cas exposé ci-après).

### 6.3.3 Compensation et codage global par région

Le schéma de principe précédent n'engendre pas de coût de représentation en régions supplémentaire, mais a le désavantage de ne pas offrir une progressivité spatiale totale. En effet, la carte de segmentation de l'image n'est rendue disponible qu'après le codage au niveau blocs. Une solution réellement progressive consisterait à décoder dans un premier temps la séquence à très bas débit uniquement au niveau régions. Le corollaire est d'obtenir la représentation en régions à  $I_{n+1}$  à partir de celle disponible à  $I_n$ .

La distinction entre l'approche présentée au paragraphe précédent et celle-ci est illustrée figure 10.20. Il existe de nombreuses méthodes pour le codage différentiel de cartes de segmentation. Notre approche présentée au chapitre 2 pour le codage des formes basé sur la grille, devrait également convenir. Cependant, cette progressivité spatiale engendre un coût supplémentaire à évaluer en fonction du nombre de régions.

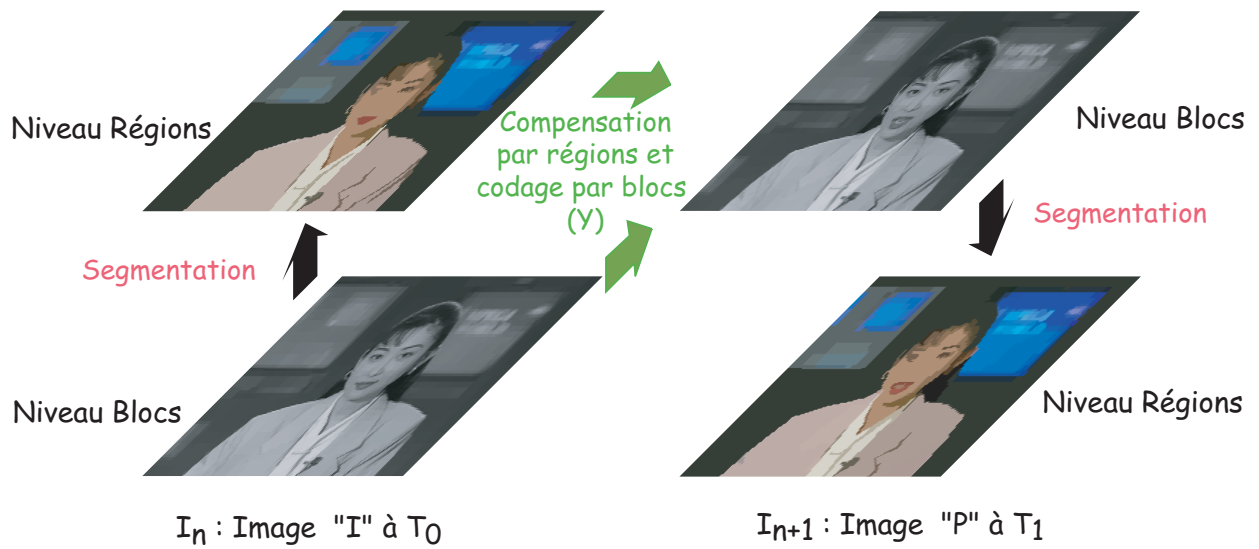
L'étape suivante réside dans le codage prédictif comme décrit précédemment, tirant éventuellement profit de la mise à jour de la carte de segmentation.

## 7 Conclusion

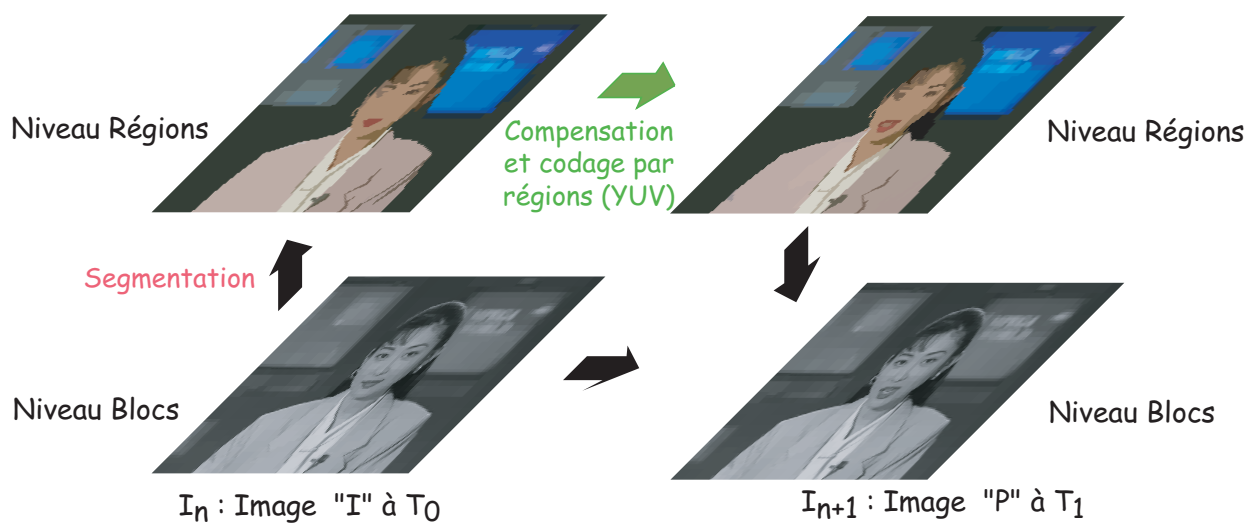
Outre un bref état-de-l'art des méthodes d'estimation de mouvement, ce chapitre comporte des pistes de réflexions concernant l'extension à la vidéo de la méthode LAR. Le comportement de la technique de segmentation introduite au chapitre 2 se révèle encourageant : les images successives d'une séquence, encodées sans compensation de mouvement (images toutes de type "I"), restent cohérentes entre elles. De ce fait, considérant ses propriétés de scalabilité présente en termes à la fois de résolution, de qualité et de régions, le LAR devient un bon candidat à l'élaboration d'une méthode efficace de codage vidéo à fortes fonctionnalités de services.

Si nous n'avons pas eu le temps de mettre en œuvre ces idées, au cours de cette thèse, nous espérons à très courts termes obtenir une première évaluation des techniques soumises au paragraphe 6.3.

Quelle que soit la stratégie choisie, les résultats d'une segmentation étant parfois arbitraires, il s'agit de s'assurer de la similitude entre deux segmentations successives.



a. Schéma à progressivité réduite - Coût des régions minimal



b. Schéma à forte progressivité - Surcoût des régions

FIG. 10.20: Schéma de codage vidéo à progressivité différenciée



## Conclusion





# Conclusion et perspectives

Le dénominateur commun de l'ensemble des travaux de ce manuscrit concerne une méthode de codage des images nommée LAR (Locally Adaptive Resolution).

L'objectif majeur de cette thèse était tout d'abord de fournir pour le traitement des images fixes de nouvelles fonctionnalités au schéma de base, à savoir :

- scalabilité de l'ensemble des étapes de compression,
- possibilité de réalisation d'une compression sans perte,
- intégration d'une segmentation à des fins de codage "intelligent".

Lors de l'exécution de ces travaux, des collaborations avec d'autres équipes de recherche nous ont conduit à envisager l'utilisation du schéma LAR d'une part comme un outil destiné à de plus larges applications (télémédecine et sécurisation de la transmission), et d'autre part comme une méthode de compression directement implantable dans des systèmes embarqués via des méthodes évoluées de prototypage rapide.

Enfin, un dernier objectif visé concernait l'extension du LAR à la vidéo.

La section 1 revient sur chacun des points précédemment évoqués afin d'en faire un bilan objectif. La section 2 présente quant à elle une série de perspectives de recherche, relevées suite aux travaux réalisés au cours de cette thèse.

## 1 Synthèse des recherches effectuées lors du travail de thèse

### 1.1 Représentation en régions

La définition d'un processus de segmentation tire parti de la représentation en blocs de taille variable propre au LAR. Si l'algorithme résultant n'est pas optimal, l'adéquation naturelle entre forme et contenu de l'image lui confère d'excellentes propriétés. Les résultats visuels obtenus attestent d'une grande qualité pour un coût de codage minimal.

Certes, une grande partie du travail est due aux recherches menées par Olivier Déforges : nous avons cependant introduit en commun la notion de VOP (Video Object Plane) dans notre codeur, concept directement dérivé du codage par régions contraintes par un masque préalablement défini. Par ailleurs, la collaboration que nous avons pu entamer, (et écourtée indépendamment de notre volonté) avec Darian Muresan (Cornell University), a permis d'améliorer plus encore la qualité des images reconstruites par un post-traitement efficace.

### 1.2 Codage sans perte des images fixes

La réalisation d'une compression sans perte s'est effectuée conjointement à l'intégration de la notion de scalabilité, par l'intermédiaire de méthodes pyramidales. Associés à une phase de

prédiction, trois codeurs différents répondant à ces exigences ont vu le jour : le LAR-APP, l'Interleaved S+P et le RWHT+P.

### 1.2.1 *Le LAR-APP*

Le LAR-APP (Approche Pyramidale Prédictive) se fonde sur l'exploitation d'un contexte de prédiction enrichi obtenu par un parcours original des niveaux de la pyramide construite. Ainsi, la décomposition de la pyramide nécessite deux descentes successives : la première encode l'image LAR basse résolution, et la seconde la texture. En outre, par niveau de la pyramide, il existe deux parcours entrelacés des pixels : le contexte de prédiction lors de la deuxième passe s'avère alors suffisamment raffiné pour réduire l'entropie des erreurs de prédiction commises. Les résultats de compression sans perte apparaissent proches de ceux obtenus par les codeurs de l'état de l'art (S+P et CALIC). Le désavantage majeur de la méthode tient dans le fait qu'elle opère entièrement dans le domaine spatial : la prédiction de la seconde diagonale s'avère peu performante, et demeure un frein à l'élaboration d'un schéma plus efficace.

La définition du LAR-APP a permis en outre de jeter les bases de toute construction pyramidale s'appuyant sur les propriétés intrinsèques du LAR. En particulier, la séparation des lois entropiques selon le niveau de la pyramide considéré et la nature des symboles à transmettre (symbole codant l'image LAR basse résolution ou la texture, distinction entre erreur de prédiction obtenue lors de la première passe et erreur calculée au cours de la seconde passe de l'algorithme). Cette répartition a priori des symboles codant l'image apparaît comme le facteur clé de réussite des solutions multirésolutions du LAR.

### 1.2.2 *Interleaved S+P*

L'interleaved S+P est issu de la réflexion suivante : le LAR-APP reste limité par les propriétés du domaine spatial dans lequel il opère. L'introduction de la transformée en S dans le schéma de codage apporte la décorrélation supplémentaire nécessaire aux données. Une solution élégante se construit alors par l'entrelacement de deux pyramides de coefficients transformés. En association avec un schéma prédictif adapté, les résultats dépassent largement les méthodes de l'état-de-l'art, qu'elles soient progressives ou non. En particulier, pour les 130 images médicales testées, les performances sont étonnantes : l'algorithme produit un encodage supérieur en moyenne à CALIC de l'ordre de 0.28 bpp ! Il en est de même pour les images composites et les images aériennes. De nouveau, la séparation des coefficients selon leurs différents critères (nature, première ou deuxième pyramide, première ou deuxième descente) apporte un gain très significatif en compression.

### 1.2.3 *Méthode RWHT+P*

Le RWHT+P introduit une solution originale, exploitant le système à décomposition différenciée initialement construit pour le LAR-APP. La technique est fondée sur la définition d'une transformée WHT 2D réversible. La conception d'une phase d'estimation, employée également à des fins d'interpolation, contribue à l'obtention d'une méthode efficace. De plus, la réalisation d'une classification de contexte dans l'espace transformé contribue à l'amélioration de l'entropie globale. Associée à la séparation des lois probabilistes (voir l'Interleaved S+P),

cette solution procure des résultats surpassant largement encore l'Interleaved S+P : l'algorithme RWHT+P se situe par la suite au delà des méthodes de l'état-de-l'art sur le plan de la compression sans perte.

Ainsi, Interleaved S+P et RWHT+P apparaissent comme des techniques alliant performances entropiques pour du codage sans perte et fonctionnalités avancées du type : scalabilité en termes de résolution, de qualité et de régions (compression localement sans perte par exemple).

### 1.3 Transmission sécurisée d'images : application à la télémédecine

En collaboration avec le laboratoire IRCCyN de Nantes, nous nous sommes intéressés au problème de transmission sécurisée d'images sur des réseaux à bas débits du type Internet. Le procédé de sécurisation doit s'entendre au sens de la protection contre la perte de paquets IP. Pour ce faire, un codage conjoint source-canal réalisé par l'association du codage LAR et de la transformée Mojette (transformée de Radon discrète et exacte) entreprend de répondre à l'objectif suivant : offrir une protection différenciée intégrant la nature hiérarchique des flux issus des méthodes multirésolution du LAR pour une qualité de service exécutée de bout-en-bout.

Dans ce contexte, l'observation de la sensibilité au bruit de l'Interleaved S+P a révélé une propriété extrêmement intéressante : la méthode s'avère robuste aux erreurs de transmission réalisées sur les flux codant les erreurs de prédiction. Dans le cadre de la télémédecine, cette qualité apparaît comme un atout. En effet, du fait de la faible propagation spatiale de l'erreur de transmission, il est toujours possible de restituer la majeure partie de l'image sans altération notable.

Ainsi, un codage conjoint source-canal efficace a été mis en œuvre de telle sorte qu'il puisse être intégré à une solution complète de télémédecine. La collaboration s'est trouvée ainsi renforcée entre les deux laboratoires, et, parallèlement, nous nous sommes impliqués dans l'organisation de journées thématiques du GdR STIC Santé.

### 1.4 Méthodologie AAA et prototypage rapide

Dans le cadre de recherches opérées sur la méthodologie AAA (Adéquation Algorithme Architecture) au sein de l'équipe Architecture du Groupe Image de l'IETR, nous avons entrepris un travail conjoint visant à l'implantation automatique des codeurs LAR sur des architectures parallèles hétérogènes multi-composants. Par le biais de la description des algorithmes sous le logiciel SynDEx, il nous a été possible en particulier de réaliser le prototypage de l'Interleaved S+P sur des plate-formes multi-DSP et multi-PC.

Sur un Pentium 4 à 2.8 GHz, le codeur et le décodeur respectent largement la contrainte temps-réel généralement appliquée à la vidéo (traitement d'une image CIF en moins de 40 ms) : cette performance s'avère d'autant plus intéressante que l'algorithme effectue un encodage scalable sans perte.

Le prototypage automatique de l'application sur des plate-formes mono-PC / mono-DSP ou encore mono-PC / bi-DSP a permis le chronométrage de l'application globale. Sur DSP du type C64x, la principale limitation de l'implantation réside dans le fait que l'ensemble des opérations ne peuvent tenir toutes en mémoire interne du DSP. Les résultats demeurent cependant très en-

courageants : moyennant quelques opérations d’optimisation, nous sommes convaincus de pouvoir obtenir un codeur et un décodeur temps-réel (sur topologie mono-DSP ou multi-DSP) totalement embarqués.

Grâce à ce travail transversal entre deux thématiques du groupe Image du laboratoire (équipes ”AAA” et ”Compression et Représentation des Images”), nous avons renforcé une collaboration durable, efficace et bénéfique pour chacun d’entre nous. Outre les avancées scientifiques évidentes opérées, nous avons également contribué à l’introduction d’une méthode commune de développement.

## 1.5 Extension du LAR à la vidéo

La dernière partie de cette thèse a traité de l’extension du schéma LAR à la vidéo : compte-tenu des avancées notables que nous avons réalisées dans les domaines de la compression d’images fixes sans perte et de la transmission sécurisée des images compressées notamment, cette étude n’a pu être menée à son terme. Notre travail se situe à un niveau prospectif permettant d’évaluer nos idées quant au système vidéo à concevoir. Ainsi, il semble judicieux d’utiliser le processus de segmentation à des fins d’encodage à très bas débits. Nous avons pu vérifier le comportement de cette segmentation au cours de l’encodage d’une séquence d’images : la cohérence temporelle du procédé s’avère suffisante pour envisager clairement son intégration à un codec vidéo.

## 2 Perspectives

De nombreux thèmes ont été abordés tout au long de cette thèse. L’élargissement du sujet général (compression d’images par la méthode LAR) a permis de soulever certaines questions et problèmes non encore résolus. Pour chaque point évoqué ci-dessus, il est ainsi possible de définir de nouvelles pistes de recherche. Dans cette section, nous essayons d’évaluer les ”restes à faire”, mettant en particulier l’accent sur les collaborations à venir.

### 2.1 Compression des images fixes avec pertes

Nous avons vu lors de la description du schéma LAR et de ses variantes la difficulté à établir une comparaison objective de la qualité des images produites. Les mesures classiques du type PSNR s’avèrent totalement inefficaces pour reproduire la perception subjective des observateurs. Ainsi, la seule évaluation correcte des images reconstruites après encodage a été obtenue en 2002 grâce aux travaux de Patrick Le Callet (IRCCyN). Depuis, de nouvelles techniques ont été introduites : il serait intéressant de pouvoir juger de la qualité des images encodées avec pertes produites par l’Interleaved S+P (images intermédiaires) ou par l’algorithme dédié aux images couleurs (exploitation de la segmentation). Pour ce faire, il est d’ores et déjà prévu d’utiliser les métriques décrites dans la référence [CB03], en collaboration avec le laboratoire de Nantes.

## 2.2 Compression des images fixes sans perte

Les excellents résultats obtenus par les algorithmes Interleaved S+P et plus encore par le RWHT+P nous amènent à penser que leur exploitation dans le domaine médical ou encore pour des images composites pourrait être poursuivie. Il reste toutefois quelques expérimentations à mener afin de réellement optimiser les performances.

Ainsi, il nous faut examiner la question suivante : l'introduction pour l'Interleaved S+P de la classification décrite pour le RWHT+P apporterait-elle aussi un gain entropique substantiel ? Si tel était le cas, il deviendrait fort probable que les taux de compression avoisineraient ceux obtenus par le RWHT+P.

Par ailleurs, l'étude de la méthode RWHT+P nécessite quelques approfondissements : au vu des résultats de l'Interleaved S+P, le RWHT+P devrait révéler des performances très intéressantes sur les images médicales, composites et aériennes. De toutes premières expérimentations en ce sens confirment notre intuition.

## 2.3 Télémédecine et sécurisation des données

Dans le cadre d'une transmission sécurisée d'images compressées par le LAR, nous avons évalué la robustesse de l'Interleaved S+P aux erreurs, en dehors de tout codage entropique. Un codeur entropique, essentiel à toute chaîne de compression complète, nécessite d'être intégré à notre système. Par la suite, une réflexion doit être menée pour définir un codage adapté et robuste. Plusieurs solutions sont envisagées. D'une part, des mécanismes de resynchronisation peuvent être introduits afin d'éviter de perdre l'ensemble des flux si une erreur commise empêche toute reconstruction ultérieure. D'autre part, un code entropique symétrique doit être ajouté afin de limiter la propagation d'une erreur de transmission sur l'ensemble d'un flux. Une solution (simple) consisterait à rendre symétrique le codeur Golomb-Rice déjà implanté. Cette perspective doit être en particulier étudiée par Shan Wang dans le cadre d'une thèse réalisée au laboratoire SIC (Poitiers) et en collaboration avec notre équipe.

Le procédé de sécurisation que nous avons mis en place au moyen de la transformée Mojette traite de la protection des paquets IP. La suite de ces travaux élaborés conjointement avec Benoît Parrein (IRCCyN) concerne l'utilisation de l'Interleaved S+P (au lieu du LAR-APP tout d'abord implanté) et la mise en œuvre d'un algorithme de protection différenciée par la prise en compte automatique de la qualité de l'image reconstruite.

Enfin, la notion de sécurité peut s'étendre également aux domaines de la cryptographie et du tatouage. En exploitant correctement les propriétés inhérentes au schéma Interleaved S+P, ces deux solutions peuvent être réalisées au moyen de formes évoluées de la transformée Mojette. Une thèse a débuté cette année (Jean Motsch-Pivette) en co-tutelle avec le laboratoire de Nantes, et vise à l'élaboration d'une solution de sécurisation des images 2D ou des volumes d'images compressées par le LAR.

## 2.4 Méthodologie AAA et prototypage rapide

Nous avons déjà exprimé au chapitre 9 les quelques soucis que nous avons rencontrés lors de l'implantation de l'Interleaved S+P sur des plate-formes multi-DSP. De ces limitations, nous avons

retiré les observations suivantes. D'une part, il reste un travail d'optimisation du code à effectuer afin de faire tenir entièrement le codeur et le décodeur en mémoire interne des DSP à disposition. Si cette solution s'avère difficile à réaliser, des recherches en cours sur l'implantation d'un mécanisme de mémoire cache pourraient être à défaut utilisées, pour des performances intéressantes. D'autre part, l'amélioration de la description SynDEx, par une définition du graphe d'algorithme à plus fine granularité, devrait permettre, du fait de l'augmentation du parallélisme potentiel ainsi exprimé, une implantation automatique multi-processeur plus proche de l'optimalité.

Par ailleurs, nous prévoyons le prototypage de l'application sur des plate-formes matérielles autres que celles utilisées (utilisation des DSP C6416, de fréquence horloge 1 GHz notamment).

A plus long terme, nous espérons pouvoir réaliser la description et l'implantation automatique des solutions de codage vidéo : si le graphe SynDEx et l'application associée existent, le système comporte des imperfections qu'il faut absolument résorber. En particulier, des allocations dynamiques liées à la conception des graphes d'adjacence (segmentation) subsistent et constituent un blocage au développement plus poussé du codec video sur cibles mixtes multi-PC et multi-DSP.

## 2.5 Le LAR et la vidéo

Lors de la description de l'extension du LAR à la vidéo, nous avons principalement indiqué les solutions que nous envisageons d'implanter (chapitre 10). L'exploitation de la technique de segmentation à des fins d'encodage à très bas débits semble être une voie prometteuse.

Par ailleurs, un projet concernant le suivi d'objets présents dans une vidéo doit voir le jour très prochainement (en attente d'une thèse). Le système imaginé doit faire le lien, via une représentation en régions efficace, entre application visant à la manipulation des objets d'une scène et compression à qualité localement variable. Une des pistes intéressantes concerne les approches dites hybrides : celles-ci exploitent les avantages des approches région et contours en considérant d'abord les objets comme une entité, puis en suivant les différentes parties les composant. Une telle solution s'inspirerait de la méthode décrite par Cavarallo et al. [CSE05]

Enfin, dans le cadre du traitement des séquences d'images médicales (du type échographie ou encore prise de vue lors d'une opération chirurgicale), il devient nécessaire de disposer d'un système de compression vidéo sans perte. Cette perspective suppose encore un travail préalable afin de définir une stratégie de codage.

Comme je l'espère, le lecteur aura pu apprécier un bilan conséquent aux perspectives non moins ambitieuses... Toutefois, l'ampleur de ces perspectives ne doit pas lui faire croire à un travail inachevé, mais plutôt à une étude dont les différentes spécifications progressives laisseraient ouvertes de nombreuses potentialités semblables à celles que peut présenter le standard MPEG-4.

# Annexe





# Annexe I

## Images médicales testées

Afin de comprendre les résultats obtenus par notre algorithme Interleaved S+P (chapitre 6), il est important d'observer les images testées. En effet, si les images naturelles testées ("lena", "baboon", "cafe", "hotel"... ) constituent un ensemble connu d'images de référence, il n'existe pas d'équivalent dans le domaine de l'image médicale. En raison de la diversité importante des techniques utilisées pour l'acquisition des données cliniques, et du peu de publications réalisées en compression d'images pure, il est difficile de présenter des résultats sans connaître l'image source traitée. Ainsi, cette annexe propose un panel non exhaustif (110 images ont été testées!) mais représentatif des images médicales mentionnées au chapitre 6.

Les images sont regroupées par type d'examen, à savoir : radiographies conventionnelle, échographies, coupes tomographiques, images par résonance magnétique et mammographies. Exceptées "angiot", "colon", et "us", toutes ces images sont issues de l'une des deux bases de données suivantes :

- <http://eelmpo.cityu.edu.hk/imagedb/>
- <http://www.cis.rit.edu/htbooks/mri/inside.htm>

Que leurs auteurs soient remerciés de la mise à disposition de leur données cliniques.

### 1 Radiographie conventionnelle



FIG. I.1: cr -  $1744 \times 2048$



FIG. I.2: x\_ray -  $2048 \times 1680$

## 2 Echographie

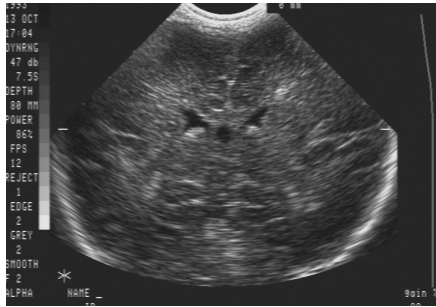


FIG. I.3: echo -  $720 \times 496$

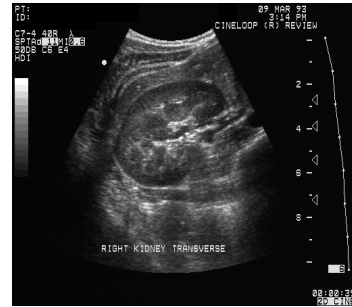


FIG. I.4: us -  $512 \times 448$

## 3 Coupes tomographiques



FIG. I.5: ct\_abdomen\_fr -  $512 \times 512$

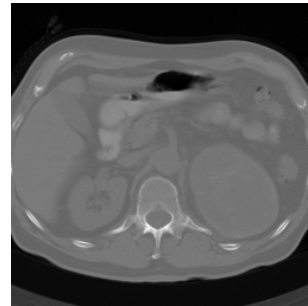


FIG. I.6: ct\_abdomen\_tr -  $256 \times 256$

## 4 Images par résonance magnétique

### 4.1 Angiographies



FIG. I.7: angio -  $512 \times 512$

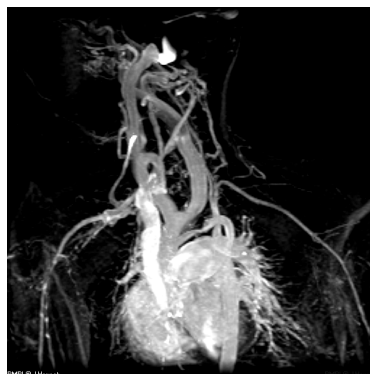


FIG. I.8: an01 -  $512 \times 512$



FIG. I.9: an02 -  $512 \times 512$

## 4.2 Tête



FIG. I.10: mr\_head\_p -  $256 \times 256$



FIG. I.11: hd01 -  $512 \times 512$

## 4.3 Autres parties du corps



FIG. I.12: ex03 -  $512 \times 512$  -  
Genou

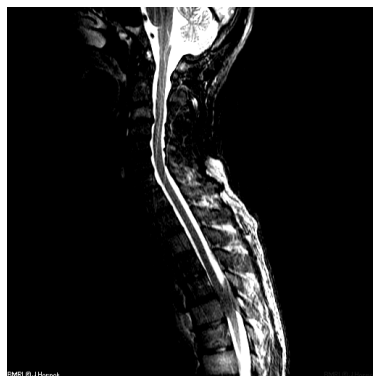


FIG. I.13: nk02 -  $512 \times 512$  -  
Cou et moelle épinière

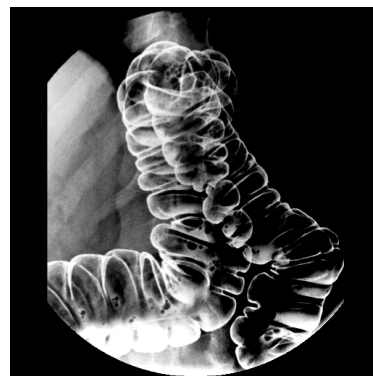


FIG. I.14: colon -  $512 \times 512$  -  
Colon



FIG. I.15: sp09 -  $512 \times 512$  -  
Colonne vertébrale

## 5 Mammographies

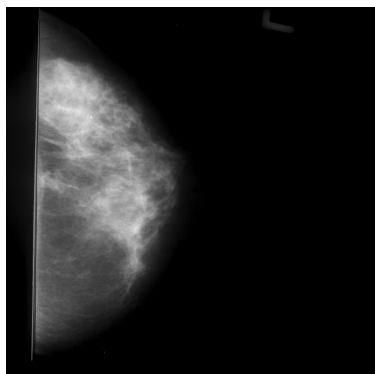


FIG. I.16: mammo -  $2048 \times 2048$

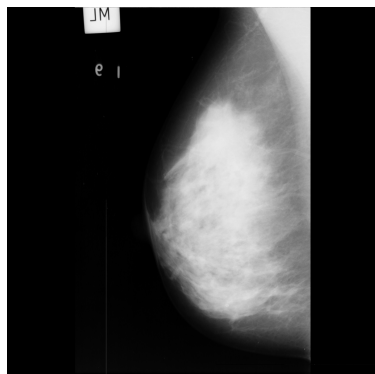


FIG. I.17: mdb003 -  $1024 \times 1024$  - Tissu glandulaire dense, aspect normal

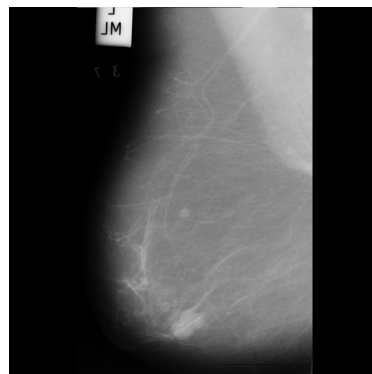


FIG. I.18: mdb005 -  $1024 \times 1024$  - Sein gras, lésion bénigne

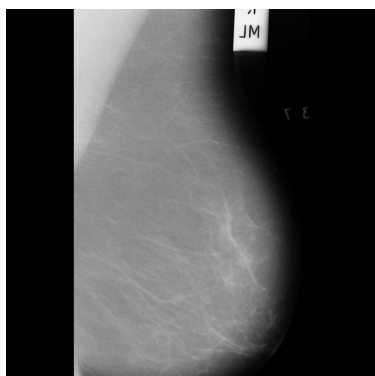


FIG. I.19: mdb006 -  $1024 \times 1024$  - Sein gras, aspect normal

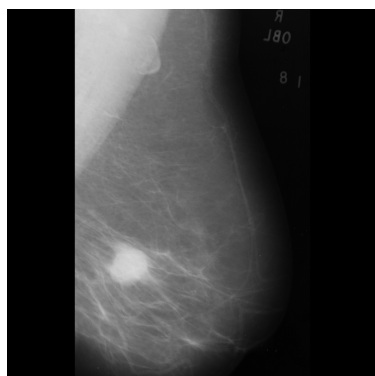


FIG. I.20: mdb028 -  $1024 \times 1024$  - Sein gras, lésion maligne

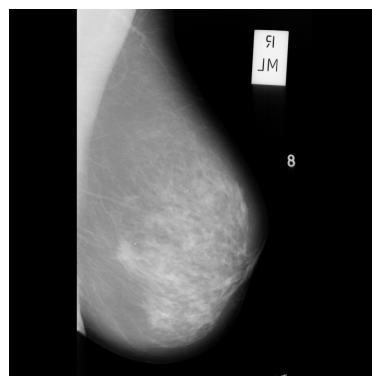


FIG. I.21: mdb058 -  $1024 \times 1024$  - Tissu glandulaire dense, lésion maligne

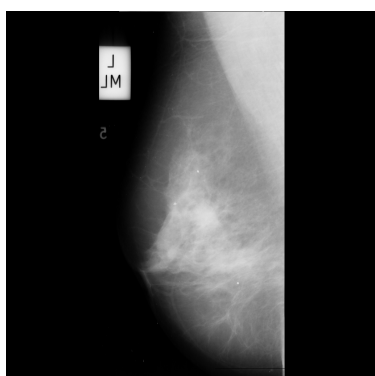


FIG. I.22: mdb063 -  $1024 \times 1024$  - Tissu glandulaire dense, lésion bénigne

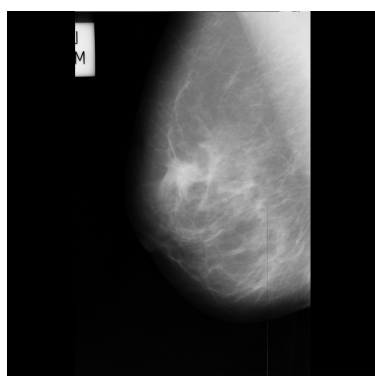


FIG. I.23: mdb117 -  $1024 \times 1024$  - Tissu glandulaire gras, lésion maligne

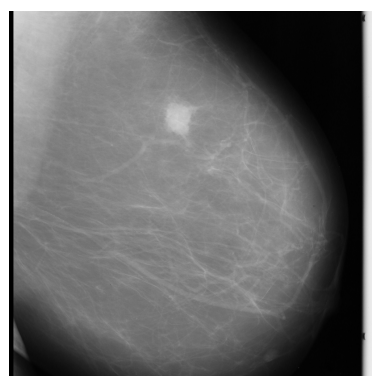


FIG. I.24: mdb134  $1024 \times 1024$  Sein gras, lésion maligne

## Annexe II

# Télémédecine : plate-forme de démonstration

L'interactivité s'avère nécessaire pour certaines applications de codage/décodage, entre autres pour celles manipulant des régions d'intérêt de façon non automatique. Nous avons donc entrepris de créer un outil intégrant une interface utilisateur évoluée, et faisant preuve d'une grande flexibilité quant aux phases de codage/décodage entrelacées. L'objectif des travaux s'inscrit dans le cadre du GdR STIC Santé, à savoir : proposer une plate-forme performante et viable à long terme, destinée à la visualisation et à la transmission via les réseaux grand public et bas débit d'images médicales compressées par la méthode LAR.

Ce travail a été réalisé suite au portage de l'application LAR-APP sous le logiciel SynDEx (voir chapitre 9) : la trame du codeur (et du décodeur associé) a été automatiquement généré par l'outil pré-cité. Deux stagiaires consécutifs ont œuvré à la réalisation de la plate-forme de démonstration de la méthode pyramidale du LAR : Christian Bossé (Master Recherche STI, Université de Rennes 1, et Projet de Fin d'Etudes, INSA de Rennes) et Olivier Milet (Projet de Fin d'Etudes, INSA de Rennes) [Bos03, Mil04]. C'est une application MFC (Microsoft Foundation Class) programmée en Visual C++, et compatible avec les systèmes d'exploitation Windows 98, 2000, NT et XP. Un projet actuellement en cours porte sur le développement d'un logiciel analogue à celui présenté dans cette annexe, mais implanté sous Dev C++ (incluant GCC, compilateur GNU multi plates-formes).

Cette annexe n'a pas pour objectif de présenter l'ensemble des choix relatifs à l'architecture logicielle de l'application, que le lecteur pourra trouver dans [Bos03]. Nous présentons ici simplement les diverses caractéristiques de notre interface.

## 1 Manipulation des fichiers et fonctions de visualisation

La plate-forme permet l'ouverture et la sauvegarde des images en niveaux de gris (8 bits) sous les formats ".bmp", ".pgm", ".gif", ".ras", ainsi que sous la forme compressée par le LAR (".lar"). L'interprétation des images médicales par un praticien requiert la réalisation de traitements spécifiques, comme la modulation de luminosité et de contraste, ou encore le zoom (figure II.1).

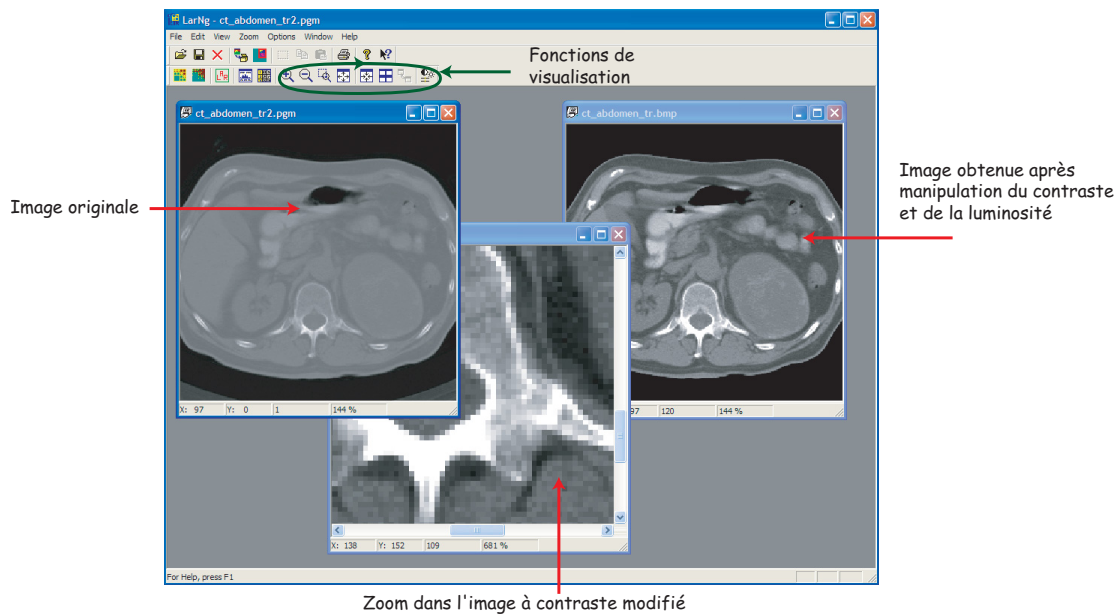


FIG. II.1: Fonctions de visualisation de la plate-forme.

## 2 Compression par le LAR-APP

Pour cette plate-forme de démonstration, dont le développement a débuté en 2003, le LAR-APP a été implanté. Des boîtes de dialogues dédiées permettent de régler à volonté les paramètres d'encodage de la source (figure II.2). Un codeur arithmétique adaptatif est ici introduit : même si son implantation n'est pas optimale, la mise en œuvre de ce codage entropique permet d'avoir une estimation de l'entropie réelle de l'encodage source proposé.

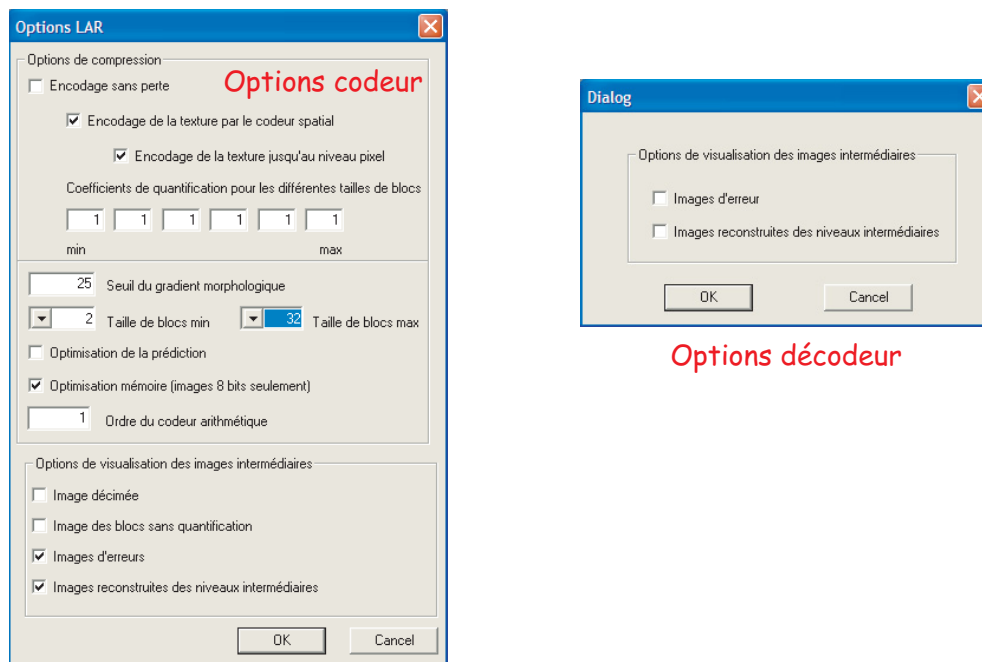


FIG. II.2: Options de compression et de visualisation des images compressées au codeur et au décodeur.

Si l'outil est d'abord conçu pour le personnel médical, il a également pour but d'offrir un certain nombre de fonctionnalités illustrant et analysant les résultats de compression : projection de la grille de partitionnement sur les images, affichage des images intermédiaires, histogramme des images et résultats de compression, fenêtre d'affichage des valeurs des pixels (figure II.3).

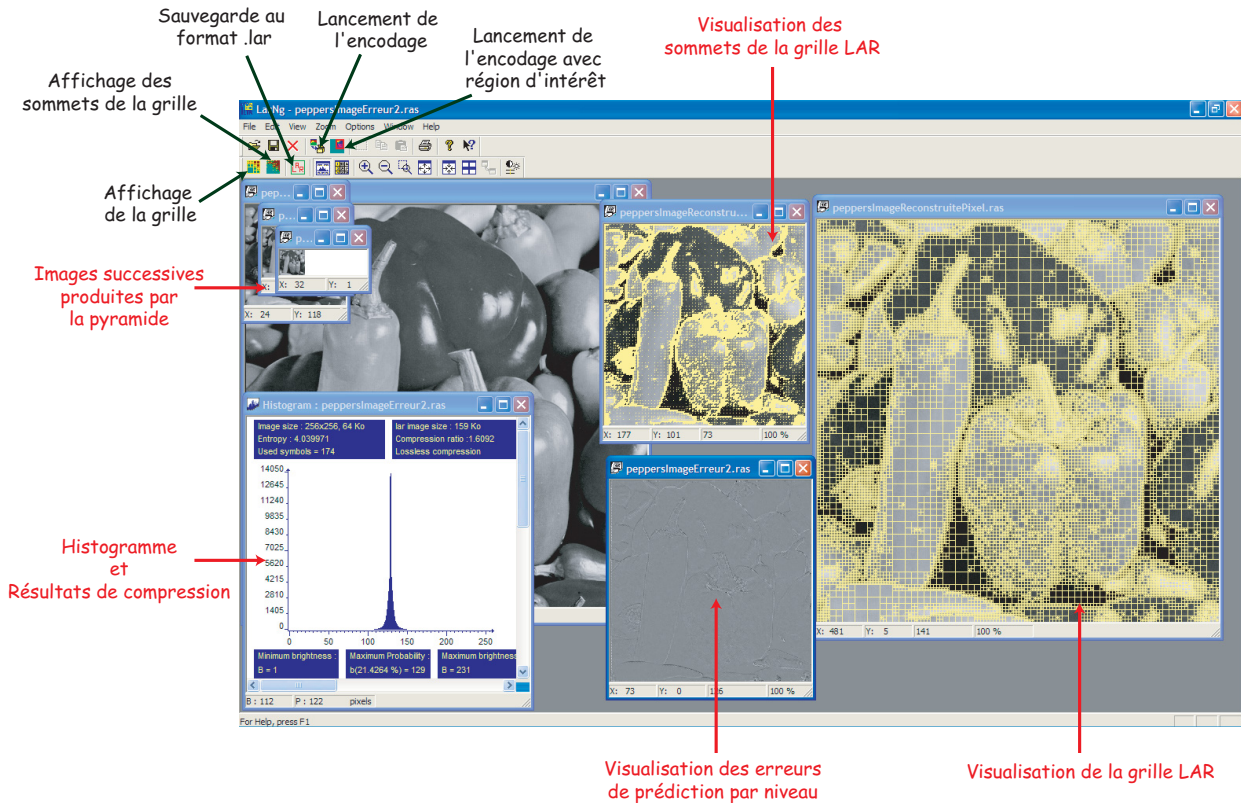


FIG. II.3: Lancement d'une compression et résultats associés.

Une commande permet l'encodage de l'image après définition d'une région d'intérêt décrite au codeur. Ainsi la grille s'adapte automatiquement à cette contrainte. La première descente de la pyramide permet alors la reconstruction de l'image  $LR^{APP}$  basse résolution complète. La seconde descente, quant à elle, ne prend en compte que les pixels situés à l'intérieur de cette région d'intérêt. Cette compression localement sans perte améliore la vitesse de transmission de l'information importante.

### 3 Perspectives

Nous avons en projet, outre le portage de la plate-forme dans un environnement compilable sous Windows ou Linux, de réaliser une application client-serveur complète, assurant une Qualité de Service de bout-en-bout. Des développements s'avèrent donc nécessaires : création d'une base de données, intégration d'un protocole de communication, introduction de la transformée Mojette-Dirac (voir chapitre 8), possibilité de définition d'une région d'intérêt au décodeur, visualisation progressive à la demande... Ce projet ambitieux devait se concrétiser cette année : le stage proposé n'a malheureusement pas été pourvu, et le travail reste pour le moment en suspens.





# Lexique

AAA	Adéquation Algorithme Architecture
BMA	Block Matching
bpb	Bit Per Block
bpp	Bit Per Pixel
CABAC	Context Adaptive Binary Arithmetic Coding
CALIC	Context-based Adaptive Lossless Image Coding
Codec	COdeur-DECodeur
DCT	Discrete Cosinus Transform
DFD	Displaced Frame Difference
DICOM	Digital Imaging and Communication in Medicine
DMA	Direct Memory Access
DSP	Digital Signal Processor
DWT	Discrete Wavelet Transform
ECMA	Equation de Contrainte du Mouvement Apparent
EEP	Equal Error Protection
EZW	Embedded Zerotree Wavelet
FEC	Forward Error Correcting
FIFO	First In First Out
FPGA	Field Programmable Gate Array
H264	Standard de compression video équivalent à MPEG-4 AVC
HBMA	Hierarchical Block Matching
Interleaved S+P	S+P entrelacé
IP	Internet Protocol
IRM	Image par Résonance Magnétique
JPEG	Joint Photographic Experts Group
LAR	Locally Adaptive Resolution
LAR-APP	LAR- Approche Pyramidale Prédictive
LOCO	LOW complexity Context-based compression
LR	Low Resolution image
MAD	Mean Absolute Difference
MICD	Modulation par Impulsions à Codage Différentiel
MPEG	Moving Picture Experts Group
MSE	Mean Square Error
PACS	Picture Archival and Communication Systems
PCI	Peripheral Component Interconnect
PDA	Personal Digital Assistant
PSNR	Peak Signal to Noise Ratio
QoS	Quality of Service
QP	Quadtree Partition
RAG	Region Adjacency Graph
RAM	Random Access Memory
ROI	Region Of Interest

RWHT	Reversible Walsh Hadamard Transform
RWHT+P	Reversible Walsh Hadamard Transform + Prédiction
S+P	Transformée en S et Prédiction
SAM	Sequential Access Memory
SBC	Subband Coding
SDB	Sundance Digital Buses
SPIHT	Set partitioning In Hierarchical Trees
SVH	Système Visuel Humain
SynDEx	Synchronized Distributed Execution
TCP/IP	Transmission Control Protocol / Internet Protocol
UEP	Unequal Error Protection
VOL	Video Object Layer
VOP	Virtual Object Plan
Webcam	Web Camera
WHT	Walsh-Hadamard Transform

# Table des figures

1.1	Schéma global LAR à deux couches : codeurs spatial et spectral . . . . .	10
1.2	Schéma de principe du codeur spatial . . . . .	11
	a. Image source Lena $512 \times 512$ - 8 bpp . . . . .	16
	b. Partition Quadtree : 13888 blocs . . . . .	16
	c. Image reconstruite basse résolution : 0.2 bits/pixel. Taux de compression : 40, PSNR 29.2 dB . . . . .	16
	d. Image reconstruite après post-traitement sur $P^{[16...2]}$ , PSNR 29.7 db . . . . .	16
1.3	Résultats pour une partition $P^{[16...2]}$ , $Th = 30$ . . . . .	16
1.4	Schéma de principe du codeur spectral pour une partition $QP^{[16...2]}$ . . . . .	18
1.5	Principe de progressivité sémantique . . . . .	19
	a. JPEG : partition $P^{[8]}$ . . . . .	19
	b. LAR : partition $P^{[16...2]}$ . . . . .	19
1.6	Répartition de l'énergie : comparaison entre partitions à taille fixe et taille variable . . . . .	19
	a. Image Lena, Cr/Cb 16 bpp . . . . .	20
	b. Cr+Cb codées blocs : 0.063 bpp . . . . .	20
	c. Image baboon, Cr/Cb 16 bpp . . . . .	20
	b. Cr+Cb codées blocs : 0.226 bpp . . . . .	20
1.7	Exemples d'images reconstruites avec codage des chrominances par le codeur spatial (Y source non modifiée) . . . . .	20
1.8	Résultats tests comparatifs de perception visuelle . . . . .	22
1.9	Résultats comparatifs de qualité visuelle . . . . .	23
	a. Image Baboon, Cr/Cb 16 bpp . . . . .	33
	b. 671 régions . . . . .	33
	c. Cr+Cb codées région : 0.020 bpp . . . . .	33
	d. Image Lena, Cr/Cb 16 bpp . . . . .	33
	e. 308 régions . . . . .	33
	e. Cr+Cb codées région : 0.009 bpp . . . . .	33
2.1	Exemples d'images reconstruites avec codage des chrominances par une représentation en régions . . . . .	33
2.2	Schéma bloc du codeur LAR couleur fondé sur une segmentation supervisée . . . . .	34
2.3	Résultats de segmentation en fonction des paramètres $Th_{Cost}$ et $CoefChrom$ sur image "Barbara". . . . .	37
2.4	Résultats de segmentation sur "Baboon", "peppers", "Fruits" et "Lena" ( $CoefChrom = 0.5$ ). . . . .	38

2.5	Codage des chrominances par régions . . . . .	39
2.6	Codage par ROI sur image “Akiyo” avec contraintes de chrominance et masque binaire. . . . .	40
3.1	Transmission d’un codage <i>non-embedded</i> . . . . .	44
3.2	Transmission d’un codage <i>embedded</i> . . . . .	44
3.3	Visualisation des différents types possibles de scalabilité pour un flux de données compressées. . . . .	45
3.4	Scalabilité temporelle pour MPEG : un exemple. Image I : image codée intra (opérations effectuées sur l’image seule) - Image P : image codée par prédiction temporelle ”forward” - Image B : image codée par prédiction temporelle bidirectionnelle. . . . .	46
3.5	Sous-espaces vectoriels d’approximation emboîtés. . . . .	47
3.6	Représentation par algorithme pyramidal. . . . .	51
3.7	Sous-espaces vectoriels de détails emboîtés. . . . .	51
3.8	Schéma-bloc d’une transformation ondelettes à 2 niveaux (signal 1D) . . . . .	52
3.9	Algorithme à Trous. . . . .	53
3.10	Visualisation des différences de grilles issues d’une décomposition multirésolution et d’une décomposition en paquets d’ondelettes. . . . .	54
3.11	Construction de la pyramide laplacienne pour un signal 1D . . . . .	56
3.12	Décomposition pyramidale en ondelettes pour un signal 2D par un codeur à deux couches. . . . .	58
3.13	Banc de filtre 1D et sa structure lifting équivalente . . . . .	59
3.14	Reconstruction en lifting d’un signal 1D . . . . .	60
3.15	Décomposition pyramidale morphologique. L’opération $OC_B$ représente l’ouverture-fermeture par l’élément structurant $B$ . La décomposition produit une image d’approximation $y_L$ où $L$ correspond au niveau le plus haut de la pyramide, et plusieurs images résiduelles $\{R_i\}_{i=1}^{L-1}$ . . . . .	62
4.1	Approximations successives d’un coefficient d’amplitude $ c $ , dans l’intervalle $[0, T]$ , où $T$ est un entier, puissance de deux. . . . .	67
4.2	Ordre de parcours des sous-bandes. . . . .	67
4.3	Filiation entre les sous-bandes. . . . .	68
4.4	Décomposition en ondelettes de CREW : alternance de la transformée en S et la transformée TS. . . . .	74
4.5	Ordre d’encodage des pixels par la méthode HINT. . . . .	75
4.6	JPEG 2000 : Prétraitement et transformée. . . . .	77
4.7	JPEG 2000 : ”Code-Block” et ”Precinct”. . . . .	78
5.1	Construction des moyennes de diagonales à coder. . . . .	82
5.2	Première passe de l’algorithme de Wu. . . . .	83
5.3	Deuxième et troisième passes de l’algorithme de Wu. . . . .	84
5.4	Schéma général du codeur LAR : deux couches de codage, avec deux types de codeur pour la texture. . . . .	85
5.5	Décomposition pyramidale et compression sans perte : images successives obtenues. . . . .	85

5.6	Indices $i$ et $j$ .	87
5.7	Construction <i>bottom-up</i> .	88
5.8	Construction incomplète de la pyramide.	89
5.9	Décomposition Top-Down de la pyramide.	90
5.10	Visualisation de l'influence de la valeur seuil $Th$ sur la compression sans perte d'une image.	94
5.11	Effet de l'interpolation sur l'image des blocs LAR (image basse résolution - 0.31 bpp) : décomposition selon la première descente jusqu'au niveau 1 pour la luminance, 2 pour les deux composantes de chrominance	96
5.12	Comparaison des images obtenues après compression à taux égal par les algorithmes JPEG, JPEG-2000 et LAR-APP.	96
5.13	Comparaison des images obtenues après compression à taux égal (0.123 bpp) par les algorithmes Intra-MPEG4, JPEG-2000, H26L (mode d'optimisation maximale) et LAR-APP.	97
6.1	Application particulière de la transformée en S sur une image pleine résolution.	100
6.2	Pyramides entrelacées : principe de construction.	101
7.1	Ensemble de configurations de blocs frontière considérés	121
	a.	122
	b.	122
7.2	Image intermédiaire (niveau $l = 3$ ) sous-échantillonnée (a) et interpolée (b) - Décomposition globale (0.089 bpp)	122
	a.	123
	b.	123
7.3	Exemple d'image intermédiaire sous-échantillonnée (a) et interpolée (b), pour une décomposition non régulière - Première descente pyramide, $l=1$ : 0.39 bpp.	123
	a. Première descente de la pyramide - $l_{min} = 5$ (1 flux) : 0.005 bpp	126
	b. Première descente de la pyramide - $l_{min} = 3$ (3 flux) : 0.073 bpp	126
	c. Première descente de la pyramide - $l_{min} = 1$ (5 flux) : 0.433 bpp	126
	d. Première descente de la pyramide - $l_{min} = 0$ (6 flux) : 1.194 bpp	126
	e. Deuxième descente de la pyramide - $l_{min} = 1$ (10 flux) : 2.014 bpp	126
	f. Deuxième descente de la pyramide - $l_{min} = 0$ (11 flux) : 3.87 bpp	126
7.4	Codage progressif sans perte	126
8.1	Visualisation des différents services réalisés par le PACS, ainsi que son intégration dans un système de soin complet.	133
8.2	Les 3 étapes d'encodage de la grille valant pour une partition Quadtree autorisant les blocs de taille $2 \times 2$ à $16 \times 16$ .	136
	A1. crosses : image originale	136
	B1. lena : image originale	136
	C1. colon : image originale	136
	A2. crosses : grille associée	136
	B2. lena : grille associée	136
	C2. colon : grille associée	136

8.3	Images utilisées et grilles originales associées . . . . .	136
	A1. crosses : grille modifiée . . . . .	138
	B1. lena : grille modifiée . . . . .	138
	C1. colon : grille modifiée . . . . .	138
	A2. crosses : image reconstruite, $PSNR = 13.58$ dB . . . . .	138
	B2. lena : image reconstruite, $PSNR = 5.22$ dB . . . . .	138
	C2. colon : image reconstruite, $PSNR = 14.99$ dB . . . . .	138
8.4	Première étape du codage de grille : remplacement d'un bloc $16 \times 16$ par un bloc $2 \times 2$ . . . . .	138
	A1. crosses : grille modifiée . . . . .	138
	B1. lena : grille modifiée . . . . .	138
	C1. colon : grille modifiée . . . . .	138
	A2. crosses : image reconstruite, $PSNR = 13.88$ dB . . . . .	138
	B2. lena : image reconstruite, $PSNR = 18.47$ dB . . . . .	138
	C2. colon : image reconstruite, $PSNR = 15.19$ dB . . . . .	138
8.5	Première étape du codage de grille : remplacement d'un bloc $2 \times 2$ , $4 \times 4$ ou $8 \times 8$ par un bloc $16 \times 16$ d'un bloc plus petit. . . . .	138
	A1. crosses : grille modifiée . . . . .	139
	B1. lena : grille modifiée . . . . .	139
	C1. colon : grille modifiée . . . . .	139
	A2. crosses : image reconstruite, $PSNR = 15.28$ dB . . . . .	139
	B2. lena : image reconstruite, $PSNR = 21.60$ dB . . . . .	139
	C2. colon : image reconstruite, $PSNR = 17.55$ dB . . . . .	139
8.6	Deuxième étape du codage de grille : remplacement d'un bloc $8 \times 8$ par un bloc $2 \times 2$ . . . . .	139
	A1. crosses : grille modifiée . . . . .	139
	B1. lena : grille modifiée . . . . .	139
	C1. colon : grille modifiée . . . . .	139
	A2. crosses : image reconstruite, $PSNR = 17.42$ dB . . . . .	139
	B2. lena : image reconstruite, $PSNR = 25.52$ dB . . . . .	139
	C2. colon : image reconstruite, $PSNR = 21.77$ dB . . . . .	139
8.7	Troisième étape du codage de grille : remplacement d'un bloc $2 \times 2$ par un bloc $4 \times 4$ . . . . .	139
	a. crosses . . . . .	140
	b. lena . . . . .	140
	c. colon . . . . .	140
8.8	Grille inconnue au décodeur : hypothèse d'une grille uniforme (blocs $16 \times 16$ ). . . . .	140
8.9	Analyse théorique de la propagation intra-niveau des erreurs . . . . .	141
	A1. crosses : Image originale . . . . .	143
	A2. crosses : Image reconstruite . . . . .	143
	A3. crosses : propagation de l'erreur . . . . .	143
	B1. lena : Image originale . . . . .	143
	B2. lena : Image reconstruite . . . . .	143
	B3. lena : propagation de l'erreur . . . . .	143
	C1. ct_abdomen_fr : Image originale . . . . .	143

C2. ct_abdomen_fr : Image reconstruite . . . . .	143
C3. ct_abdomen_fr : propagation de l'erreur . . . . .	143
D1. colon : Image originale . . . . .	143
D2. colon : Image reconstruite . . . . .	143
D3. colon : propagation de l'erreur . . . . .	143
8.10 Valeur d'erreur de prédiction erronée observée sur le pixel $Y_{l_{max}}(0,0) = Y_4(0,0)$ . . .	143
a. Image niveau 4 au codeur . . . . .	144
b. Image niveau 4 au décodeur . . . . .	144
c. Propagation de l'erreur au niveau 4 . . . . .	144
8.11 Propagation de l'erreur intra-niveau (image crosses, niveau $l = l_{max} = 4$ ). Erreur sur la première valeur transmise. . . . .	144
A1. lena : Image reconstruite . . . . .	145
B1. abdomen : Image reconstruite . . . . .	145
C1. colon : Image reconstruite . . . . .	145
A2. lena : propagation de l'erreur . . . . .	145
B2. abdomen : propagation de l'erreur . . . . .	145
C2. colon : propagation de l'erreur . . . . .	145
8.12 Valeur d'erreur de prédiction erronée observée sur un pixel du niveau $l_{max} = 4$ . . .	145
A1. Erreur sur $z_1^{3,2}$ , gradient de la 1 <sup>ère</sup> pyramide en S du niveau 3 . . . . .	145
B1. Erreur sur $z_0^{3,3}$ , moyenne de la 2 <sup>nde</sup> pyramide en S du niveau 3 . . . . .	145
C1. Erreur sur $z_1^{3,3}$ , gradient de la 2 <sup>nde</sup> pyramide en S du niveau 3 . . . . .	145
A2. Propagation de l'erreur issue de A1. . . . .	145
B2. Propagation de l'erreur issue de B1. . . . .	145
C2. Propagation de l'erreur issue de C1. . . . .	145
8.13 Observation de la propagation intra-niveau de l'erreur. . . . .	145
A1. Erreur sur $z_1^{3,2}$ . . . . .	146
B1. Erreur sur $z_0^{3,3}$ . . . . .	146
C1. Erreur sur $z_1^{3,3}$ . . . . .	146
A2. Propagation de l'erreur issue de A1. . . . .	146
B2. Propagation de l'erreur issue de B1. . . . .	146
C2. Propagation de l'erreur issue de C1. . . . .	146
8.14 Observation de la propagation complète de l'erreur (intra- et inter-niveau) appliquée sur l'image LAR basse résolution : décomposition totale de la pyramide. . . . .	146
A1. Erreur sur $z_1^{3,2}$ . . . . .	146
B1. Erreur sur $z_0^{3,3}$ . . . . .	146
C1. Erreur sur $z_1^{3,3}$ . . . . .	146
A2. Propagation de l'erreur issue de A1. . . . .	146
B2. Propagation de l'erreur issue de B1. . . . .	146
C2. Propagation de l'erreur issue de C1. . . . .	146
8.15 Observation de la propagation complète de l'erreur (intra- et inter-niveau) appliquée sur l'information de texture : décomposition totale de la pyramide. . . . .	146
a. Faible erreur sur $z_1^{3,3}$ - image LAR . . . . .	147
b. Erreur sur $z_1^{1,2}$ - image LAR . . . . .	147



c. Erreur sur $z_1^{1,2}$ - texture . . . . .	147
d. Erreur sur $z_1^{3,2}$ - image LAR. . . . .	147
e. Erreur sur $z_1^{1,2}$ - image LAR. . . . .	147
f. Erreur sur $z_1^{1,2}$ - texture. . . . .	147
d. Erreur sur $z_1^{3,2}$ - image LAR. . . . .	147
e. Erreur sur $z_1^{1,2}$ - image LAR. . . . .	147
f. Erreur sur $z_1^{1,2}$ - texture. . . . .	147
8.16 Exemple d'images reconstruites après réception d'une valeur erronée. . . . .	147
8.17 Transformée Mojette directe : construction des projections. . . . .	149
8.18 Exemple d'application de la transformée Mojette selon les directions (1, 0), (1, 2) et (-2, 1) . . . . .	149
8.19 Schéma à trois niveaux de protections avec séparation en mémoires tampon géométriques. La construction d'un paquet est réalisée par la concaténation de projections du même angle. $p_{a,b}$ représente la $b^{me}$ projection du $a^{me}$ buffer géométrique. . . . .	152
8.20 Représentation des paquets IP : visualisation de la taille des paquets, nombre de projections nécessaires (en rouge) et nombre de projections redondantes (en bleu) par sous-flux. Image "angioma" traitée par le LAR-APP. . . . .	154
8.21 Qualité de l'image reconstruite pour différent taux de réception. 100% signifie que toutes les projections (tous les paquets) ont été reçues. . . . .	155
9.1 Principe général de SynDEx . . . . .	160
9.2 Environnement SynDEx . . . . .	161
9.3 Processus complet de prototypage . . . . .	164
9.4 Répétition implicite d'une opération (mécanisme de Fork / Join) . . . . .	165
9.5 Graphe correspondant au niveau le plus haut de la hiérarchie SynDEx (Interleaved S+P et LAR-APP). Visualisation des éléments de description élémentaire d'une application. . . . .	167
9.6 Description commune à l'Interleaved S+P et au LAR-APP. . . . .	168
9.7 Construction et compression de la grille. . . . .	169
9.8 Visualisation des différentes étapes de la décomposition pyramidale. . . . .	170
9.9 Application vidéo du LAR et segmentation. . . . .	171
9.10 Graphe d'architecture pour application mono-PC (pentium). . . . .	172
9.11 Graphe d'architecture pour application client-serveur. . . . .	172
9.12 Description SynDEx de l'encodage des erreurs de prédiction et de la mise en forme des flux pour l'Interleaved S+P. . . . .	174
9.13 Graphe d'architecture pour deux DSP C64 et un PC. . . . .	177
9.14 Architecture Sundance utilisée à des fins d'encodage video performant. . . . .	178
9.15 Graphe d'architecture pour un DSP C64 et un PC. . . . .	178
10.1 Insertion des blocs d'estimation et de compensation de mouvement dans la chaîne de codage. . . . .	186
10.2 Projection d'un mouvement 3D sur une image plane. . . . .	186
10.3 Différence entre mouvement réel et mouvement apparent : cas de <i>l'enseigne de barbier</i> . . . . .	187
10.4 Mouvement apparent induit par une variation d'illumination. . . . .	188

10.5	Visualisation géométrique de l'équation du flot optique. Le flot optique d'un pixel donné de l'image peut être n'importe lequel des points situés sur la tangente. . . .	189
10.6	Représentation du mouvement : mouvement global, basé pixels, blocs, régions ou maillage. . . . .	191
10.7	Recherche logarithmique en 3 étapes du bloc candidat à l'appariement (Three Step Search) . . . . .	193
10.8	Principe de la descente de gradient. . . . .	194
10.9	Estimation multirésolution. . . . .	195
10.10	Sens de l'estimation : recherche de vecteurs en arrière ou en avant. . . . .	196
10.11	Partitions du macrobloc autorisées (○) ou non (×). . . . .	197
10.12	Deformable Block Matching Algorithm : principe. . . . .	198
10.13	Hierarchical Block Matching Algorithm : principe. . . . .	199
10.14	Schéma-blocs d'un codeur simple. . . . .	201
10.15	Combinaison des approches hiérarchiques et multirésolution. . . . .	201
10.16	Segmentation itérative - suivi temporel. . . . .	202
10.17	Schéma de réalisation du codeur VideoQ. . . . .	204
a.	81 régions (U : 0.004 bpp, V : 0.006 bpp, Contrôle : 0.059 bpp) . . . . .	207
b.	88 régions (U : 0.004 bpp, V : 0.006 bpp, Contrôle : 0.059 bpp) . . . . .	207
c.	83 régions (U : 0.006 bpp, V : 0.006 bpp, Contrôle : 0.059 bpp) . . . . .	207
d.	77 régions (U : 0.005 bpp, V : 0.006 bpp, Contrôle : 0.059 bpp) . . . . .	207
e.	85 régions (U : 0.004 bpp, V : 0.006 bpp, Contrôle : 0.060 bpp) . . . . .	207
f.	86 régions (U : 0.006 bpp, V : 0.006 bpp, Contrôle : 0.059 bpp) . . . . .	207
g.	78 régions (U : 0.004 bpp, V : 0.006 bpp, Contrôle : 0.061 bpp) . . . . .	207
h.	89 régions (U : 0.004 bpp, V : 0.006 bpp, Contrôle : 0.060 bpp) . . . . .	207
i.	93 régions (U : 0.004 bpp, V : 0.007 bpp, Contrôle : 0.062 bpp) . . . . .	207
j.	86 régions (U : 0.005 bpp, V : 0.006 bpp, Contrôle : 0.062 bpp) . . . . .	207
k.	100 régions (U : 0.005 bpp, V : 0.007 bpp, Contrôle : 0.063 bpp) . . . . .	207
l.	91 régions (U : 0.005 bpp, V : 0.006 bpp, Contrôle : 0.063 bpp) . . . . .	207
10.18	Etude de la segmentation par contrôle de la chrominance sur les 12 premières images de la séquence "Foreman_cif.yuv" : observation de la représentation en régions. . .	207
a.	81 régions, 0.277 bpp (Grille : 0.045 bpp, Y : 0.163 bpp, U : 0.004 bpp, V : 0.006 bpp, Contrôle : 0.059 bpp) . . . . .	208
b.	88 régions, 0.282 bpp (Grille : 0.045 bpp, Y : 0.166 bpp, U : 0.004 bpp, V : 0.006 bpp, Contrôle : 0.059 bpp) . . . . .	208
c.	83 régions, 0.281 bpp (Grille : 0.045 bpp, Y : 0.165 bpp, U : 0.006 bpp, V : 0.006 bpp, Contrôle : 0.059 bpp) . . . . .	208
d.	77 régions, 0.278 bpp (Grille : 0.044 bpp, Y : 0.163 bpp, U : 0.005 bpp, V : 0.006 bpp, Contrôle : 0.059 bpp) . . . . .	208
e.	85 régions, 0.281 bpp (Grille : 0.045 bpp, Y : 0.165 bpp, U : 0.004 bpp, V : 0.006 bpp, Contrôle : 0.060 bpp) . . . . .	208
f.	86 régions, 0.281 bpp (Grille : 0.045 bpp, Y : 0.164 bpp, U : 0.006 bpp, V : 0.006 bpp, Contrôle : 0.059 bpp) . . . . .	208

g. 78 régions, 0.281 bpp (Grille : 0.045 bpp, Y : 0.165 bpp, U : 0.004 bpp, V : 0.006 bpp, Contrôle : 0.061 bpp)	208
h. 89 régions, 0.283 bpp (Grille : 0.046 bpp, Y : 0.167 bpp, U : 0.004 bpp, V : 0.006 bpp, Contrôle : 0.060 bpp)	208
i. 93 régions, 0.289 bpp (Grille : 0.046 bpp, Y : 0.170 bpp, U : 0.004 bpp, V : 0.007 bpp, Contrôle : 0.062 bpp)	208
j. 86 régions, 0.290 bpp (Grille : 0.046 bpp, Y : 0.170 bpp, U : 0.005 bpp, V : 0.006 bpp, Contrôle : 0.062 bpp)	208
k. 100 régions, 0.294 bpp (Grille : 0.047 bpp, Y : 0.171 bpp, U : 0.005 bpp, V : 0.007 bpp, Contrôle : 0.063 bpp)	208
l. 91 régions, 0.298 bpp (Grille : 0.047 bpp, Y : 0.176 bpp, U : 0.005 bpp, V : 0.006 bpp, Contrôle : 0.063 bpp)	208
10.19 Etude de la segmentation par contrôle de la chrominance sur les 12 premières images de la séquence "Foreman_cif.yuv".	208
a. Schéma à progressivité réduite - Coût des régions minimal	211
b. Schéma à forte progressivité - Surcoût des régions	211
10.20 Schéma de codage vidéo à progressivité différenciée	211
I.1 cr - $1744 \times 2048$	223
I.2 x_ray - $2048 \times 1680$	223
I.3 echo - $720 \times 496$	224
I.4 us - $512 \times 448$	224
I.5 ct_abdomen_fr - $512 \times 512$	224
I.6 ct_abdomen_tr - $256 \times 256$	224
I.7 angio - $512 \times 512$	224
I.8 an01 - $512 \times 512$	224
I.9 an02 - $512 \times 512$	224
I.10 mr_head_p - $256 \times 256$	225
I.11 hd01 - $512 \times 512$	225
I.12 ex03 - $512 \times 512$ - Genou	225
I.13 nk02 - $512 \times 512$ - Cou et moelle épinière	225
I.14 colon - $512 \times 512$ - Colon	225
I.15 sp09 - $512 \times 512$ - Colonne vertébrale	225
I.16 mammo - $2048 \times 2048$	226
I.17 mdb003 - $1024 \times 1024$ - Tissu glandulaire dense, aspect normal	226
I.18 mdb005 - $1024 \times 1024$ - Sein grassex, lésion bénigne	226
I.19 mdb006 - $1024 \times 1024$ - Sein grassex, aspect normal	226
I.20 mdb028 - $1024 \times 1024$ - Sein grassex, lésion maligne	226
I.21 mdb058 - $1024 \times 1024$ - Tissu glandulaire dense, lésion maligne	226
I.22 mdb063 - $1024 \times 1024$ - Tissu glandulaire dense, lésion bénigne	226
I.23 mdb117 - $1024 \times 1024$ - Tissu glandulaire grassex, lésion maligne	226
I.24 mdb134 $1024 \times 1024$ Sein grassex, lésion maligne	226
II.1 Fonctions de visualisation de la plate-forme.	228

---

II.2 Options de compression et de visualisation des images compressées au codeur et au décodeur. . . . .	228
II.3 Lancement d'une compression et résultats associés. . . . .	229



# Liste des tableaux

1.1	Valeurs de quantification selon la taille . . . . .	15
2.1	Influence des paramètres de segmentation sur “Barbara”. . . . .	36
5.1	Codage sans perte de Lena (4.44 bpp - $T_h = 20$ ) : entropie (en bpp) par niveau de pyramide, par passe et par taille des blocs (bit de la moyenne exclu). . . . .	92
5.2	Entropie du premier ordre (bit/pixels) de la méthode Interleaved S+P (avec partition Quadtree ou non), CALIC et S+P. . . . .	95
6.1	Entropie du premier ordre de la méthode Interleaved S+P (avec partition ou non), CALIC et S+P. . . . .	105
6.2	Codage sans perte de Lena (4.31 bpp - $T_h = 20$ ) : entropie (en bpp) par niveau de pyramide, par passe, par coefficient et par taille des blocs. . . . .	106
6.3	Nombre de blocs concernés pour chaque niveau, selon la passe et la nature du coefficient (image Lena) . . . . .	106
6.4	Comparaisons entre S+P, CALIC et Interleaved S+P sur un ensemble d’images médicales de diverses natures. . . . .	108
6.5	Mammographies : tissus glandulaires denses . . . . .	109
6.6	Mammographies : seins grasieus . . . . .	110
6.7	Mammographies : tissus glandulaires grasieus . . . . .	111
6.8	Entropie du premier ordre (bit/pixels) de la méthode Interleaved S+P (avec partition Quadtree ou non), CALIC et S+P - Images composites . . . . .	112
6.9	Entropie du premier ordre (bit/pixels) de la méthode Interleaved S+P (avec partition Quadtree ou non), CALIC et S+P - Images aériennes . . . . .	112
7.1	Entropie du premier ordre (bit/pixels) de la méthode pyramidale RWHT, avec la transformée classique S+P. . . . .	119
7.2	Entropie du premier ordre (bit/pixels) de la méthode RWHT+P (mise en œuvre de la phase de prédiction), CALIC et S+P. . . . .	122
7.3	Entropie du premier ordre (bit/pixels) de la méthode RWHT+P, avec partition Quadtree (Qd) ou non, CALIC et S+P. . . . .	123
7.4	Entropie du premier ordre (bit/pixels) de la méthode RWHT+P, avec partition Quadtree (Qd) ou non, classification (Cl) ou non, CALIC et S+P. . . . .	125
7.5	Entropie du premier ordre (bit/pixels) de la méthode RWHT+P, avec partition Quadtree (Qd) ou non, classification (Cl) ou non, CALIC et S+P. . . . .	126

8.1	Taille des différents niveaux de reconstruction ( <i>layer</i> ) et le nombre de projections nécessaires et redondantes associées. Image "angioma" traitée par le LAR-APP (3.35 bpp) . . . . .	153
9.1	Relevé des temps d'exécution des principales fonctions de l'Interleaved S+P sur un PC de type Pentium 4, 2.8 GHz, pour la première image de la séquence "foreman_cif.yuv" (352×288). . . . .	173
9.2	Relevé des temps d'exécution du traitement pyramidal après suppression des phases de (dé)quantification sur un PC de type Pentium 4, 2.8 GHz, pour la première image de la séquence "foreman_cif.yuv" (352×288). . . . .	174
9.3	Relevé des temps d'encodage des erreurs de prédiction par niveau après modification, sur un PC de type Pentium 4, 2.8 GHz, pour la première image de la séquence "foreman_cif.yuv" (352×288). . . . .	175
9.4	Relevé des temps d'exécution du codec modifié sur un PC de type Pentium 4, 2.8 GHz, pour la première image de la séquence "foreman_cif.yuv" (352×288). . . . .	175
9.5	Relevé des temps d'exécution des principales fonctions de l'Interleaved S+P sur un DSP de type C64x (utilisé en mémoire interne), pour la première image de la séquence "foreman_cif.yuv" (352×288). . . . .	179
9.6	Relevé des temps d'exécution des principales fonctions de l'Interleaved S+P après modifications sur un DSP de type C64x (utilisé en mémoire interne), pour la première image de la séquence "foreman_cif.yuv" (352×288). . . . .	180

# Bibliographie

## Publications personnelles

- [Bab02] MARIE BABEL, “Etude de Schémas Prédicatifs dans des Représentations non Uniformes pour du Codage Avec et Sans Perte,” M.S. thesis, IETR Groupe Image - INSA de Rennes, Février-Juillet 2002.
- [BD03] MARIE BABEL AND OLIVIER DÉFORGES, “Décomposition Pyramidale par Prédiction et Transformée pour du Codage avec et sans Perte,” in *COmpression et REprésentation des Signaux Audiovisuels, CORESA’03*, Lyon, France, Jan. 16-17 2003.
- [BDR03a] MARIE BABEL, OLIVIER DÉFORGES, AND JOSEPH RONSIN, “Adaptive Multi-Resolution Scheme for Efficient Image Compression,” in *Picture Coding Symposium, PCS’03*, Saint-Malo, France, April 23-25 2003.
- [BDR03b] MARIE BABEL, OLIVIER DÉFORGES, AND JOSEPH RONSIN, “Décomposition pyramidale à redondance minimale pour compression d’images sans perte,” in *GRETSI’03*, Paris, France, Sept. 8-11 2003.
- [BDR03c] MARIE BABEL, OLIVIER DÉFORGES, AND JOSEPH RONSIN, “Lossless and Lossy Minimal Redundancy Pyramidal Decomposition for Scalable Image Compression Technique,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP’03*, Hong Kong, April 6-10 2003, Conference cancelled - Invited paper in ICME 2003, vol. III, pp. 249–252.
- [BDR03d] MARIE BABEL, OLIVIER DÉFORGES, AND JOSEPH RONSIN, “Lossless and Lossy Minimal Redundancy Pyramidal Decomposition for Scalable Image Compression Technique,” in *4th IEEE International Conference on Multimedia and Expos , ICME’03*, Baltimore, USA, July 6-9 2003, vol. 3, pp. 161–164.
- [BDR04a] MARIE BABEL, OLIVIER DÉFORGES, AND JOSEPH RONSIN, “Télémédecine sur Réseaux Bas Débits Sécurisés,” GDR Santé - Thème B : Signaux et Images en Santé. Journée thématique compression des signaux et des images en médecine, 30 Juin 2004.
- [BDR<sup>+</sup>04b] MARIE BABEL, OLIVIER DÉFORGES, JOSEPH RONSIN, LAURENT BÉDAT, NICOLAS NORMAND, BENOÎT PARREIN, AND JEAN-PIERRE GUÉDON, “Télémédecine sur Réseaux Bas Débits Sécurisés,” in *Colloque les Nouvelles Technologies dans la cité, TIC & Cité 2004*, Rennes, France, December 2004, pp. 93–96.
- [BDR05a] MARIE BABEL, OLIVIER DÉFORGES, AND JOSEPH RONSIN, “Interleaved S+P : Décomposition pyramidale entrelacée à contexte de prédiction enrichi,” in *GRETSI’05*, Septembre 2005, A paraître.



- [BDR05b] MARIE BABEL, OLIVIER DÉFORGES, AND JOSEPH RONSIN, “Interleaved S+P Pyramidal Decomposition with Refined Prediction Model,” in *IEEE International Conference on Image Processing, ICIP’05*, Septembre 2005, A paraître.
- [BPD<sup>+</sup>05] MARIE BABEL, BENOÎT PARREIN, OLIVIER DÉFORGES, NICOLAS NORMAND, JOSEPH RONSIN, AND JEAN-PIERRE GUÉDON, “Secured and Progressive Transmission of Compressed Images on the Internet : Application to Telemedicine,” in *SPIE 17th Annual Symposium / Electronic Imaging - Internet Imaging*, 16-20 Jan. 2005, vol. 5670, pp. 126–136.
- [DBBR04] OLIVIER DÉFORGES, MARIE BABEL, LAURENT BÉDAT, AND JOSEPH RONSIN, “Codec LAR couleur : Schéma de représentation et de compression d’images couleur basé sur l’adaptation de la résolution locale et sur la représentation en régions auto-extractibles,” *Traitement du Signal*, vol. 21, no. 5, Janvier 2004.
- [DBR03] OLIVIER DÉFORGES, MARIE BABEL, AND JOSEPH RONSIN, “Codage d’Objets de Type VOP par Représentation en Régions Auto-Extractible,” in *GRETSI’03*, Paris, France, Sept. 8-11 2003.
- [DBR<sup>+</sup>04] OLIVIER DÉFORGES, MARIE BABEL, JOSEPH RONSIN, LAURENT BÉDAT, NICOLAS NORMAND, BENOÎT PARREIN, AND JEAN-PIERRE GUÉDON, “Le LAR aux Mojettes,” in *COmpression et REprésentation des Signaux Audiovisuels, CORESA’04*, Lille, France, May 2004, pp. 165–168.
- [RBD<sup>+</sup>03] MICKAËL RAULET, MARIE BABEL, OLIVIER DÉFORGES, JEAN-FRANÇOIS NEZAN, AND YVES SOREL, “Automatic Coarse-Grain Partitioning and Automatic Code Generation for Heterogeneous Architectures,” in *IEEE Workshop on Signal Processing Systems, SIPS’03*, Seoul, Korea, August 27-29 2003, pp. 316–321.
- [ZB02] MARIUSZ ZIÓLKO AND MARIE BABEL, “Application of Wavelet-Cosine Transform to Image Compression,” in *International TISCP Workshop on Spectral Methods and Multirate Signal Processing, SMMSP’02*, Toulouse, September 7-8 2002.

## Principes généraux de codage

- [Gra97] ROBERT M. GRAY, “Signal processing and the multimedia information infrastructure,” in *First International Conference on Information, Communications, and Signal Processing*, 9-12 September 1997, Keynote Lecture.
- [KIK85] M. KUNT, A. IKONOMOPOULOS, AND M. KOCHER, “Second generation image coding techniques,” *Proceedings of the IEEE*, vol. 73, no. 4, pp. 549–575, April 1985.
- [Nic92] H. NICOLAS, *Hiérarchie de modèles de mouvement et méthodes d’estimation associées. Application au codage de séquences d’images*, Thèse de doctorat, Université de Rennes 1, September 1992.
- [Sai99] AMIR SAID, *Wavelet-Based Image Compression*, Imaging Technology Department, Hewlett Packard Laboratories, 1999.
- [TK96] L. TORRES AND M. KUNT, “Video coding : The second generation approach,” *Kluwer Academic*, 1996.

## Standards de compression (images fixes et vidéo)

- [BCM00] M. BOLIEK, C. CHRISTOPOULOS, AND E. MAJANI, “JPEG 2000,” in *Coding of Still Picture*, 2000.
- [CR03] C. NEUMANN AND V. ROCA, “A Survey of Large Scale Multimedia Streaming Techniques for the Internet,” International Workshop on Multimedia Interactive Protocols and Systems (MIPS’03), November 18-21 2003, Tutorial, [http\unskip\penalty\@M://www.inrialpes.fr/planete/people/chneuman/](http://www.inrialpes.fr/planete/people/chneuman/).
- [Gro99] MPEG-7 REQUIREMENTS GROUP, *MPEG-7 : Context, Objectives and Technical Roadmap, V.12*, ISO/IEC JTC 1/SC29/WG11 MPEG99/N2861, D.Lib Magazine, Vancouver, July 1999.
- [H264 MPEG-4 10 AVC02] H264 MPEG-4 10 AVC, *Joint Committee Draft (CD)*, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEGn 3rd Meeting : Fairfax, Virginia, USA, May 2002.
- [MGBB00] MICHAEL W. MARCELLIN, MICHAEL J. GORMISH, ALI BILGIN, AND MARTIN P. BOLIEK, “An Overview of JPEG-2000,” in *Data Compression Conference*, 2000, pp. 523–544.
- [MPEG-4 Committee00] MPEG-4 COMMITTEE, *Generic Coding of Moving Pictures and Associated Audio Information : Video*, ISO/IEC, 2000, May 2000.
- [OBL<sup>+</sup>04] JÖRN OSTERMANN, JAN BORMANS, PETER LIST, DETLEV MARPE, MATTHIAS NARROSCHE, FERNANDO PEREIRA, THOMAS STOCKHAMMER, AND THOMAS WEDI, “Video Coding with H.264/AVC : Tools, Performance, and Complexity,” *IEEE Circuits and Systems Magazine*, vol. 4, no. 1, pp. 7–28, First Quarter 2004.
- [SCE01a] A.N. SKODRAS, C.A. CHRISTOPOULOS, AND T. EBRAHIMI, “The JPEG2000 Still Image Compression Standard,” *IEEE Signal Processing*, pp. 36–58, September 2001.
- [SCE01b] A.N. SKODRAS, C.A. CHRISTOPOULOS, AND T. EBRAHIMI, “JPEG2000 : The upcoming still image compression standard,” *PRL*, vol. 22, no. 12, pp. 1337–1345, October 2001.
- [TM02] DAVID S. TAUBMAN AND MICHAEL W. MARCELLIN, “JPEG2000 : Standard for interactive imaging,” *Proceedings of the IEEE*, vol. 90, no. 8, pp. 1336–1357, August 2002.
- [Wal91] G.K. WALLACE, “The jpeg still picture compression standard,” *CACM*, vol. 34, no. 4, pp. 30–44, April 1991.

## Construction du codeur LAR simple

- [BSB97] L. BÉDAT, A. SAADANE, AND D. BARBA, “Masking Effects of Perceptual Colour Components on Achromatic Grating,” in *ECVP 97*, Helsinki, August 1997.

- [DSEO92] B. DEKNUYDT, J. SMOLDERS, L. VAN EYCKEN, AND A. OOSTERLINCK, "Color Space Choice for Nearly Reversible Image Compression," in *SPIE, Visual Communications and Image Processing '92*, Boston, November 1992, vol. 1818, pp. 1300–1311.
- [G<sup>+</sup>95] B. GANDHI ET AL., "Differential Adaptive Run Coding," Tech. Rep., ISO Working Document ISO/IEC JTC1/SC29/WG1 N204, 1995.
- [Gra58] R.E. GRAHAM, "Predictive Quantizing of Television Signals," *IREWESCON Convention Record*, vol. 22, no. 4, pp. 147–157, August 1958.
- [LM94] M. A. LOSADA AND K.T. MULLEN, "The Spatial Tuning of Chromatic Mechanisms Identified by Simultaneous Masking," *Vision Research*, vol. 34, no. 3, pp. 331–341, Februar 1994.
- [SS87] C.A. SHAFFER AND H. SAMET, "Optimal Quadtree Construction Algorithms," *Computer Vision, Graphics, Image processing*, vol. 37, no. 3, pp. 402–419, March 1987.
- [Str90] P. STROBACH, "Tree-Structured Scene Adaptive Coder," *IEEE Trans. on Communication*, vol. 38, no. 4, pp. 477–486, April 1990.
- [VWS90] K.K. DE VALOIS, M.A. WEBSTER, AND E. SWITKES, "Orientation and Spatial-Frequency Discrimination for Luminance and Chromatic Gratings," *Journal of the Optical Society of America*, vol. 7, no. 6, pp. 1034–1049, June 1990.
- [WSS96] M.J. WEINBERGER, G. SEROUSSI, AND G. SAPIRO, "LOCO-I : a low complexity, context-based, lossless image compression algorithm," *IEEE Data Compression Conference*, pp. 141–150, March 1996.

## Construction de la représentation en régions du LAR

- [BG89] J.-M BEAULIEU AND M. GOLDBERG, "Hierarchy in Picture Segmentation : a Stepwise Optimization Approach," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, no. 2, pp. 150–163, Februar 1989.
- [BSB99] L. BÉDAT, A. SAADANE, AND D. BARBA, "Optimal Quantisation of Colour Image," in *ECVP 99*, August 1999.
- [EK98] T. EBRAHIMI AND M. KUNT, "Visual Data Compression for Multimedia Applications," *Proceedings of the IEEE*, vol. 86, no. 6, pp. 1109–1125, June 1998.
- [HEMK98] K. HARIS, S.N. EFSTRATIADIS, N. MAGLAVERAS, AND A.K. KATSAGGELOS, "Hybrid Image Segmentation Using Watersheds And Fast Region Merging," *IEEE Trans. on Image Processing*, vol. 7, no. 12, pp. 1684–1699, December 1998.
- [KWH82] J. KRAUSKOPF, D.R. WILLIAMS, AND D.W. HEELEY, "The Cardinal Directions of Color Space," *Vision Research*, vol. 22, no. 9, pp. 1123–1131, 1982.
- [MAIS96] J. MAEDA, V.V. ANH, T. ISHIZAKA, AND Y. SUZUKI, "Integration of Local Fractal Dimension and Boundary Edge in Segmenting Natural Images," in *IEEE International Conference on Image Processing, ICIP'96*, 1996, vol. 1, pp. 845–848.
- [Mul85] K.T. MULLEN, "The Contrast Sensitivity of Human Colour Vision to Red-Green and Blue-Yellow Chromatic Gratings," *Journal on Physiology*, vol. 359, pp. 381–400, 1985.

- [PZ95] N. PETERFREUND AND Y.Y. ZEEVI, “Nonuniform Image Representation in Area-Of-Interest Systems,” *Trans. on Image Processing*, vol. 4, no. 9, pp. 1202–1212, September 1995.
- [SHB93] M. SONKA, V. HLAVAC, AND R. BOYLE, “Image Processing, Analysis, and Machine Vision,” in *Chapman & Hall*, 1993.
- [Tre98] A. TREMEAU, *Analyse d’Images Couleurs : du Pixel à la Scène*, H.D.R., Université de Saint-Etienne, May 1998.
- [WRK91] P. WILLEMIN, T. REED, AND M. KUNT, “Image Sequence Coding by Split and Merge,” *IEEE Trans. on Communication*, vol. 39, no. 12, pp. 1845–1855, December 1991.

## Références sur le LAR

- [BDC00] L. BÉDAT, O. DEFORGES, AND J. L. CORRE, “Low bit rate codec based on lar method for video surveillance via internet,” in *Proc. EUSIPCO*, September 2000.
- [Bos03] CHRISTIAN BOSSÉ, “Développement d’une plate-forme de transmission d’images médicales par compression à qualité localement variable,” M.S. thesis, IETR Groupe Image - INSA de Rennes, Février-Juillet 2003.
- [Déf04] OLIVIER DÉFORGES, *Codage d’images par la méthode LAR et méthodologie Adéquation Algorithme Architecture : de la définition des algorithmes de compression au prototypage rapide sur architectures parallèles hétérogènes*, Habilitation à diriger des recherches, Université de Rennes 1, Novembre 2004.
- [DR99] O. DEFORGES AND J. RONSIN, “Nonuniform Sub-Sampling using Squares Elements : a Fast Still Image Coding at Low Bit-Rate,” in *International Picture Coding Symposium PCS’99*, Portland, Oregon, April 1999.
- [DR00] O. DEFORGES AND J. RONSIN, “Region of Interest Coding for Low Bit-Rate Image Transmission,” in *Proc. International Conference on Multimedia and Expo ICME’2000*, 30 July-2 Aug 2000, vol. 1, pp. 107 – 110.
- [DR02] O. DEFORGES AND J. RONSIN, “Supervised Segmentation at Low Bit Rates for Region Representation and Color Image Compression,” in *IEEE International Conference on Multimedia and Expo, ICME’2002*, 26-29 Aug 2002, vol. 1, pp. 665 – 668.
- [Mil04] OLIVIER MILET, “Télémédecine : amélioration d’une plate-forme de codage d’images,” M.S. thesis, IETR Groupe Image - INSA de Rennes, Février-Juillet 2004.

## Décomposition en sous-bandes

- [Agh95] FARZIN AGHDASI, “Multiresolution Processing of Cloud Images,” in *World-wide Information system for Renewable Energy - Solar World Congress*, Harare, Zimbabwe, 1995.
- [ASH87] E.H. ADELSON, E.P. SIMONCELLI, AND R. HINGORANI, “Orthogonal Pyramid Transforms for Image Coding,” *SPIE*, vol. 845, pp. 50–58, October 1987.

- [BA83] PETER J. BURT AND EDWARD H. ADELSON, "The Laplacian Pyramid as a Compact Image Code," *IEEE Transactions on Communications*, vol. COM-31,4, pp. 532–540, 1983, [citeseer.nj.nec.com/burt83laplacian.html](http://citeseer.nj.nec.com/burt83laplacian.html).
- [BDBC97] K. BELLOULATA, F. DAVOINE, A. BASKURT, AND J.M. CHASSERY, "Compression Hybride Fractale/Sous-Bande Accélérée," in *Seizième Colloque GRETSI'97 sur le Traitement du Signal et des Images*, Grenoble, France, September 1997, vol. 1, pp. 287–290.
- [BF89] H. BLUME AND A. FAND, "Reversible and Irreversible Image Data Compression using the S-Transform and Lempel-Ziv Coding," in *SPIE*, 1989, vol. 1091, Medical Image III : Image capture and Display, pp. 2–18.
- [BGM<sup>+</sup>00] KRISTINA BOUCHITTE, SOUAD GUEMGHAR, OLIVIER MAGNEAU, JULIE VANDEN-BUSSCHE, AND MARYLÈNE WARTHMANN, "Projet : Bases d'ondelettes," Ecole Nationale Supérieure des Télécommunications, 2000, [http\unskip\penalty\@M://www.tsi.enst.fr/tsi/enseignement/ressources/mti/ondelettes/AMR.htm](http://www.tsi.enst.fr/tsi/enseignement/ressources/mti/ondelettes/AMR.htm).
- [Che95] Y. CHEE, "Using Fractal Coding for Progressive Image Transmission," Tech. Rep. 04-95, School of Electronics and Computer Engineering, Curtin University of Technology, Australia, 1995.
- [CM89] R. R. COIFMAN AND Y. MEYER, "Nouvelles bases orthonormées de  $L_2(r)$  ayant la structure du système de Walsh," Tech. Rep., Department of Mathematics, Yale University, 1989.
- [CM95] DIDIER CALLE AND ANNICK MONTANVERT, "Pyramides d'images résiduelles par expansion inductive," Tech. Rep. RR1995-36, Laboratoire de l'informatique et du Parallélisme - ENS Lyon, November 1995.
- [CWF76] R. E. CROCHIERE, S. A. WEBER, AND F. L. FLANAGAN, "Digital Coding of Speech in Subbands," *Bell System Technical Journal*, pp. 1069–1085, 1976.
- [Dav98] GEOFFREY M. DAVIS, "A Wavelet-Based Analysis of Fractal Image Compression," *IEEE Transactions on Image Processing*, vol. 7, no. 2, pp. 141–154, 1998, [citeseer.nj.nec.com/davis98waveletbased.html](http://citeseer.nj.nec.com/davis98waveletbased.html).
- [dM98] NICOLAS DE MAISTRE, *Progressivité et Scalabilité pour la Représentation d'Images Numériques*, Ph.D. thesis, IRISA / Université de Rennes 1, 1998.
- [EVE97] O. EGGER, V. VAERMAN, AND T. EBRAHIMI, "Subband Image Coding - An overview," Technical Report LTS 97.06, Ecole Polytechnique Fédérale de Lausanne, Department of Electrical Engineering, Signal Processing Laboratory, June 20 1997.
- [HZLL89] R. M. HARALICK, X. ZHUANG, C. LI, AND J. LEE, "The Digital Morphological Sampling Theorem," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-37, no. 12, pp. 2067–2090, December 1989.
- [JF94] BJAOUI A ., STARCK J.L., AND MURTAGH F., "Restauration des Images Multi-Echelles par l'Algorithme à Trous," *Traitement du Signal*, vol. 11, no. 3, pp. 229–243, 1994.
- [Lap02] FLORENCE LAPORTERIE, *Représentations Hiérarchiques d'Images avec des Pyramides Morphologiques. Application à l'Analyse et à la Fusion Spatio-Temporelle de Données en Observation de la Terre.*, Ph.D. thesis, INP - ENSEEIHT, Toulouse, 28 Mai 2002.

- [Mal89a] STÉPHANE MALLAT, “Multifrequency Channel Decompositions of Images and Wavelet Models,” *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 37, pp. 2091–2110, 1989.
- [Mal89b] STÉPHANE MALLAT, “A Theory for Multiresolution Signal Décomposition : the Wavelet Representation,” in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1989, vol. 2, pp. 674–693.
- [MMOP03] MICHEL MISITI, YVES MISISTI, GEORGES OPPENHEIM, AND JEAN-MICHEL POGGI, *Les Ondelettes et leurs Applications*, Hermes Sciences ed., Lavoisier, 2003.
- [NM05] MATHIEU NOWAK AND YVES MEYER, “La surprenante ascension des ondelettes,” *La Recherche*, , no. 383, pp. 56–59, Février 2005.
- [Pag03] RONAN LE PAGE, *Analyse de l’onde P d’un électrocardiogramme : application au risque de fibrillation auriculaire*, Ph.D. thesis, Université de Bretagne Ouest, 25 Février 2003.
- [SA92] EERO P. SIMONCELLI AND EDWARD H. ADELSON, “Subband Transforms,” in *Subband Image Coding*, John Woods, Ed. 1992, no. 137, pp. 143–192, MA : Kluwer Academic Publishers.
- [Sah99] SUBHASIS SAHA, “Image Compression - From DCT to Wavelets : a Review,” 1999, URL <http://my.engr.ucdavis.edu/ssaha/sahaimgcoding.html>.
- [SSB93] H. SENANE, A. SAADANE, AND D. BARBA, “The computation of visual bandwidths and their impact in image decomposition and coding,” in *International Conference and signal Processing Applications and Technology*, Santa-Clara, California, Septembre 1993, vol. 1, pp. 766–770.
- [Sta02] J.-L. STARCK, “Nonlinear Multiscale Transforms,” in *Multiscale and Multiresolution Methods*, T. Barth, T. Chan, and R. Haimes, Eds. 2002, pp. 239–278, Springer-Verlag.
- [Swe95] W. SWELDENS, “The Lifting Scheme : A New Philosophy in Biorthogonal Wavelet Constructions,” in *Wavelet Applications in Signal and Image Processing III*, A. F. Laine and M. Unser, Eds. 1995, pp. 68–79, Proc. SPIE 2569.
- [Swe96] W. SWELDENS, “Wavelets : What next ?,” *Proc. IEEE*, vol. 84, no. 4, pp. 680–685, 1996.
- [Swe98] WIM SWELDENS, “The Lifting Scheme : A Construction of Second Generation Wavelets,” *SIAM Journal on Mathematical Analysis*, vol. 29, no. 2, pp. 511–546, 1998.
- [WO86] J. WOODS AND S. O’NEIL, “Subband Coding of Images,” in *IEEE Transactions on Accoustics, Speach and Signal Processing*, October 1986, pp. 1278–1288.
- [YOY99] YOUNGJUN YOO, ANTONIO ORTEGA, AND BIN YU, “Image Subband Coding using Context-Based Classification ans Adaptive Quantization,” in *IEEE Trans. Image Processing*, Dec. 1999, vol. 8, pp. 1702–1715.

## Codage pyramidal : Etat de l’art

- [AAB96] B. AIAZZI, L. ALPARONE, AND S. BARONTI, “A Reduced Laplacian Pyramid for Lossless and Progressive Image Communication,” in *IEEE Trans. Communications*, January 1996, vol. 44, pp. 18–22.

- [AFB97] A. ABRARDO, L. ALPARONE F., AND BARTOLINI, "Encoding-Interleaved Hierarchical Interpolation for Lossless Image Compression," *SP*, vol. 56, no. 3, pp. 321–328, February 1997.
- [BB02] H. BEKKOUCHE AND M. BARRET, "Adaptive Mutiresolution Decomposition : Application to Lossless Image Compression," in *IEEE International Conference on Acoustics Speech and Signal Processing*, May 2002, vol. IV, pp. 3533–3536.
- [BBHH97] KAI UWE BARTHEL, SVEN BRANDAU, WOLFGANG HERMESMEIER, AND GUIDO HEISING, "Zerotree Wavelet Coding Using Fractal Prediction," in *Proceedings of IEEE International Conference on Image Processing (ICIP'97)*, October 1997, pp. 26–29.
- [CDSY98] A. R. CALDERBANK, I. DAUBECHIES, W. SWELDENS, AND B.-L. YEO, "Wavelet Transforms that Map Integers to Integers," *Applied and Computational Harmonic Analysis*, vol. 5, no. 3, pp. 332–369, July 1998.
- [CO96] C. CHRYSAFIS AND A. ORTEGA, "Context-based Adaptive Image Coding," in *Proc. of the 30th Asilomar Conf. on Signals, Sys. and Computers*, November 1996.
- [ERVMGU95] B. ESCALANTE-RAMIREZ, F. VENEGAS-MARTINEZ, AND F. GARCIA-UGALDE, "Pyramidal predictive image coding with polynomial transforms," in *Proceedings SPIE*, 1995, vol. 2148, pp. 107–117.
- [EY86] T. ENDOH AND Y. YAMAZAKI, "Progressive Coding Scheme for Multilevel Images," in *Proc. Picture Coding Symposium*, 1986, pp. 21–22.
- [FIBK99] S. FUKUMA, M. IWAHASHI, N. BUABTHONG, AND N. KAMBAYASHI, "Inter Color and Inter / Intra Band Prediction on Reversible Wavelet and Lossless Multi-Channel Prediction," in *IEEE International Conference on Image Processing (ICIP)*, October 1999, vol. 4, pp. 53–57.
- [Fow03] J. E. FOWLER, "Embedded Wavelet-Based Image Compression : State of the Art (Eingebettete Wavelet-basierte Bildkompression : Stand der Technik)," in *Information Technology*, September-October 2003, vol. 25, pp. 256–262.
- [GFF97] ZHU GUANGJIN, ZHAO FENGGUANG, AND JIANG FENG, "Pyramid Linear Prediction Coding for Images," *Journal of Shangai University*, vol. 1, no. 2, pp. 155–129, 1997.
- [HV92] P. G. HOWARD AND J. S. VITTER, "Error Modeling for Hierarchical Lossless Image Compression," in *Proceedings of the 1992 IEEE Data Compression Conference (DCC '92)*, March 1992, pp. 269–278.
- [HV94] P. G. HOWARD AND J. S. VITTER, "Fast Progressive Lossless Image Compression," in *Proceedings of the 1994 IST/SPIE International Symposium on Electronic Imaging Science and Technology*, February 1994.
- [HV95] D. HOULDING AND J. VALSEY, "Low Entropy Image Pyramids for Efficient Lossless Coding," in *IEEE Trans. Image Processing*, August 1995, vol. 4, pp. 1150–1153.
- [HWG97] U. HORN, T. WIEGAND, AND B. GIROD, "Bit Allocation Methods for Closed-Loop Coding of Oversampled Pyramid Decompositions," in *IEEE International Conference on Image Processing (ICIP)*, October 1997, vol. 2, pp. 17–21.

- [ICB<sup>+</sup>00] M. IWAHASHI, S. CHOKCHAITAM, N. BUABTHONG, P. ZAVARSKY, AND N. KAMBAYASHI, "Lossless Scalable Coding of Images via Lossless Multi-Channel Prediction," in *IEICE Trans. on Fundamentals*, July 2000, vol. E83-A, pp. 1450–1457.
- [IP99] A. ISLAM AND W. A. PEARLMAN, "An Embedded and Efficient Low-Complexity Hierarchical Image Coder," in *Visual Communications and Image Processing '99*, Proceedings of SPIE, Ed., January 1999, vol. 3653, pp. 294–305.
- [Jai81] A. J. JAIN, "Image Data Compression : A Review," *IEEE Proc.*, vol. 69, no. 3, pp. 349–389, March 1981.
- [JJK<sup>+</sup>97] R. L. JOSHI, H. JAFARKHANI, J. H. KASNER, T. R. FISCHER, N. FARVARDIN, M. W. MARCELLIN, AND R. H. BAMBERGER, "Comparison of Different Methods of Classification in Subband Coding of Images," in *IEEE Trans. Image Processing*, November 1997, vol. 6, pp. 1473–1486.
- [LK92] A. S. LEWIS AND G. KNOWLES, "Image Compression Using the 2-D Wavelet Transform," *IEEE Trans. on Image Processing*, vol. 1, no. 2, pp. 244–250, April 1992.
- [LK99] J. LI AND J. KUO, "Image Compression with a Hybrid Wavelet-Fractal Coder," *IEEE Trans on Image Processing*, vol. 8, no. 6, pp. 868–873, June 1999.
- [Lux77] P. LUX, "A Novel Set of Closed Orthogonal Functions for Picture Coding," *Archive für Elektronik und Übertragungstechnik*, vol. 31, no. 7, pp. 267–274, 1977.
- [MB01] M. W. MARCELLIN AND A. BILGIN, "Quantifying the Parent-Child Coding Gain in Zero-tree Based Coders," *IEEE Signal Processing Letters*, vol. 8, no. 3, pp. 67–69, March 2001.
- [MS95] N. D. MEMON AND K. SAYOOD, "Lossless image compression : a comparative study," in *Proc. SPIE (Still-Image Compression)*, vol. 2418, no. 8–20, Feb. 1995.
- [PM96] P. PISCAGLIA AND B. MACQ, "Multiresolution Lossless Compression Scheme," in *ICIP96*, 1996.
- [RC95] R. RINALDO AND G. CALVAGNO, "Image Coding by Block Prediction of Multiresolution Subimages," in *IEEE Trans. on Image Processing*, July 1995, pp. 909–920.
- [Sha93] J.M. SHAPIRO, "Embedded Image Coding using Zerotrees of Wavelet Coefficients," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.
- [SP93] A. SAID AND W. PEARLMAN, "Reversible Image Compression via Multiresolution Representation and Predictive Coding," in *SPIE Visual Communications and Image Processing Symposium*, Cambridge, November 1993, vol. 2094, pp. 664–674.
- [SP96a] AMIR SAID AND WILLIAM A. PEARLMAN, "An image multiresolution representation for lossless and lossy image compression," *IEEE Trans. on Image Processing*, vol. 5, pp. 1303–1310, September 1996.
- [SP96b] AMIR SAID AND WILLIAM A. PEARLMAN, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 6, pp. 243–250, June 1996.
- [SP96c] AMIR SAID AND WILLIAM A. PEARLMAN, "The Precedence of the Work of Said and Pearlamn over Rich's CREW," Tech. Rep., Rensselaer Polytechnic Institute, 1996.



- [SZB96] EDWARD L. SCHWARTZ, AHMAD ZANDI, AND MARTIN BOLIEK, "Implementation of Compression with Reversible Embedded Wavelets," No.22, RICOH California Research Center, [http\unskip\penalty\@M://www.ricoh.co.jp/rdc/techreport/No22/r9/r9.htm](http://www.ricoh.co.jp/rdc/techreport/No22/r9/r9.htm), July 1996.
- [Tau99] DAVID TAUBMAN, "High Performance Scalable Image Compression with EBCOT," in *IEEE Trans. on Image Processing*, 1999, vol. 9.
- [TZ94] D. TAUBMAN AND A. ZAKHOR, "Multirate 3-D Subband Coding of Video," in *IEEE Trans. Image Processing*, 1994, vol. 3, pp. 572–588.
- [VCER97] O. VIVEROS-CANCINO AND B. ESCALANTE-RAMIREZ, "Parametric image coding by means of polynomial transforms," in *Proc. SPIE, Visual Communications and Image Processing '97*, January 1997, vol. 3024, pp. 864–872.
- [Wu97] XIAOLIN WU, "High-Order Context Modeling and Embedded Conditional Entropy Coding of Wavelet Coefficients for Image Compression," in *Proc. of 31st Asilomar Conference on Signals, Systems, and Computers*, November 1997, vol. 2, pp. 1378–1382.
- [Wu01] XIAOLIN WU, *The Transform and Data Compression Handbook*, chapter Compression of Wavelet Transform Coefficients, K. R. Rao et al., Boca Ratin, CRC Press LLC, 2001.
- [ZASB95] AHMAD ZANDI, JAMES D. ALLEN, EDWARD L. SCHWARTZ, AND MARTIN BOLIEK, "CREW : Compression with Reversible Embedded Wavelets," in *Proc. of DCC*, 1995, pp. 212–221.

## Construction de l'approche pyramidale prédictive

- [DR00] O. DÉFORGES AND J. RONSIN, "Codage de Région d'Intérêt par la Méthode LAR," *CORESA'00*, Octobre 2000.
- [JCP98] H.Y. JUNG, T.Y. CHOI, AND R. PROST, "Rounding Transform and its Application for Lossless Pyramid Structured Coding," *IEEE Trans. Image Processing*, vol. 7, no. 2, pp. 234–237, Février 1998.
- [MT97] B. MEYER AND P. TISHER, "TMW : a new Method for Lossless Image Compression," *Proc. Picture Coding Symposium, Berlin*, pp. 533–538, 1997.
- [WMS95] X. WU, N. MEMON, AND K. SAYOOD, "A Context-based, Adaptive, Lossless/Nearly-Lossless Coding Scheme for Continuous-Tone Images (CALIC)," 1995.
- [WSS96] M.J. WEINBERGER, G. SEROUSSI, AND G. SAPIRO, "Loco-i : a low complexity, context-based, lossless image compression algorithm," *roc. Of the IEEE Data Compression Conference*, pp. 141–150, 1996.
- [Wu97] X. WU, "Lossless Compression of Continuous-Tone Images, via Context Selection, Quantization and Modelling," *IEEE Trans. on Image Processing*, vol. 6, no. 5, pp. 656–664, May 1997.

## Interpolation par *Optimal Recovery*

- [Are89] V. V. ARESTOV, “Optimal Recovery of Operators and Related Problems,” in *Proc. Steklov Mathematical Institute*, 1989, vol. 189, pp. 3–20.
- [GW59] M. GOLOMB AND H. F. WEINBERGER, “Optimal Approximation and Error Bounds,” in *On Numerical Approximation*, Maddison, 1959, pp. 117–190, R. E. Langer ed., The University of Wisconsin Press.
- [MP01] D.D. MURESAN AND T.W. PARKS, “Optimal Recovery Approach to Image Interpolation,” in *ICIP’01*, October 2001, vol. III, pp. 848–851.
- [MP05] D. D. MURESAN AND T. W. PARKS, “Demosaicing Using Optimal Recovery,” *IEEE Transactions on Image Processing*, vol. 14, no. 2, pp. 267–278, Februar 2005.
- [MR76] C.A. MICHELLI AND T.J. RIVLIN, “A survey of optimal recovery,” in *Optimal Estimation in Approximation Theory*, New York, 1976, p. 154, Plenum Press.
- [Mur02] DARIAN MURESAN, “Review of Optimal Recovery,” Tech. Rep. TR-2002-10, DMMD, 2002.
- [SP92] R.G. SHENOY AND T. W. PARKS, “An Optimal Recovery Approach to Interpolation,” *IEEE Transactions on Signal Processing*, vol. 40, no. 8, pp. 1987–1996, August 1992.

## Construction de la RWHT+P

- [Abh02] GURUGE CHARITH K. ABHAYARATNE, *Lossless and Nearly Lossless Digital Video Coding*, Ph.D. thesis, University of Bath, United Kingdom, 2002.
- [KS01] K. KOMATSU AND K. SEZAKI, “Lossless 2D discrete Walsh-Hadamard Transform,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP’01*, May 2001, vol. 3, pp. 1917 – 1920.
- [PD97] W. PHILIPS AND K. DENECKER, “A lossless version of the Hadamard transform,” in *Proceedings of the ProRISC workshop on circuits, systems and signal processing*, 11 1997, pp. 443–450.

## PACS, télémedecine et téléradiologie

- [AKP<sup>+</sup>04] C. ANTONIADES, E. KYRIACOU, C. PATTICHIS, A. KOUPPIS, AND C. SCHIZAS, “Emergency Telemedicine Infrastructure Using Mobile Communication Links,” in *7th World Congress on Injury Prevention and Safety Promotion*, Vienne, Juin 2004.
- [Boz03] CHRISTEL LE BOZEC, *Systèmes d’information des plateaux techniques d’imagerie (PACS). Démarche IHE*, Maîtrise SBM Informatique Médicale, Décembre 2003.
- [DCO99] W.T. DEJARNETTE, D. CSIPO, AND A. ORTH, *A Standards Based Distributed PACS Architecture*, Février 1999.
- [FC00] LI FAN AND CHANG WEN CHEN, *Multimedia Image and Video Processing*, chapter Telemedicine : a Multimedia Communication Perspective, pp. 531–542, CRC Press, ling guan, sun-yuan kung and jan larsen edition, 2000.

- [FHH04] GUY FRIJA, PHILIPPE HALIMI, AND ANNE HERNIGOU, *Réseau d'images de l'Hôpital Georges Pompidou*, Hôpital Européen Georges Pompidou - Formation continue, 2004.
- [FHR<sup>+</sup>04] GUY FRIJA, ANNE HERNIGOU, MARIE-PIERRE REVEL, LILIANE PALMENTIER, FRANÇOIS MENENT, CENDRINE LABORDE, PIERRE BOIRON, MARIE-HÉLÈNE GENET, AND PHILIPPE HALIMI, *Facteurs d'efficacité du PACS*, Hôpital Européen Georges Pompidou - Formation continue, 2004.
- [Gib00] BERNARD GIBAUD, "Interopérabilité syntaxique et sémantique pour l'échange de données et de connaissances en imagerie médicale," Thèse d'habilitation à diriger des recherches, Université de Rennes 1, 2000.
- [Gre05] HENRIJ ECHTERNACH GREGERSEN, "Diagnostic Reading of Radiology Images on a wireless PDA : Fact or Fiction ? Futuristic or Foolish ?," in *ECR'05*, Février 2005.
- [Hoc03] SALEM HOCINI, "Télémédecine, aspects techniques et organisationnels," 26-27 Juin 2003.
- [Med97] MICHAEL M. MEDENIS, "Security in Teleradiology Systems : Requirements and Proposed Mechanisms," M.S. thesis, Electrical and Computer Engineering Department, University of Arizona, Mai 1997.
- [NSHV04] H. NAZERAN, S. SETTY, E. HALTIWANGER, AND V.GONZALEZ, "A PDA-based flexible telecommunication system for telemedicine applications," in *26th Annual International Conference of the Engineering in Medicine and Biology Society, EMBC'04*, Septembre 2004, vol. 1, pp. 2200–2203.
- [Pet04] ANDRÉ PETITET, "URGENCES...mieux les assurer....mieux les suivre...la TELEMEDECINE...Quel apport?...Quelle amélioration ?...," in *Colloque les Nouvelles Technologies dans la cité, TIC & Cité 2004*, Rennes, France, Decembre 2004, pp. 51–59.
- [PKV<sup>+</sup>02] C.S. PATTICHIS, E. KYRIACOU, S. VOSKARIDES, M.S. PATTICHIS, R.ISTEPANIAN, AND C.N. SCHIZAS, "Wireless Telemedicine Systems : An Overview," *IEEE Antennas and Propagation Magazine*, vol. 44, no. 2, pp. 143–153, 2002.
- [PS04] PEDER C. PEDERSEN AND DALYS SEBASTIAN, "Wireless Technology Applications in a Rural Hospital," in *2004 ATA Annual Meeting*, Mai 2004.
- [TKO96] M. TSIKNAKIS, D.G. KATEHAKIS, AND S.C. ORPHANOUDAKIS, "Intelligent Image Management in a Distributed PACS and Telemedicine Environment," *IEEE Communications Magazine*, vol. 34, no. 7, pp. 36–45, 1996.
- [TO01] JEFFREY TEITSWORTH AND ALAN ORTH, "Pragmatic Teleradiology : One Administrator's Trial by Fire - Empirical Design of a Teleradiology System," in *ATA Annual Meeting*, Juin 2001.
- [Vuc03] JELEND VUCETIC, "Telemedicine : The Future of Wireless Applications," in *Southeast Wireless'03*, Decembre 2003.

## Transformée Mojette

- [Aut02] FLORENT AUTRUSSEAU, *Tatouage d'images fondé sur la modélisation du Système Visuel Humain et sur la transformation Mojette*, Thèse de doctorat, Université de Nantes et Ecole Centrale de Nantes, Novembre 2002.

- [Edi04] JPEG EDITORS, “ISO/IEC CD 15444-11 / ITU-T SG8 JPEG 2000 image coding system - Part 11 : Wireless JPEG2000 Committee Draft,” 2004.
- [GN03] JEAN-PIERRE GUÉDON AND NICOLAS NORMAND, “Multimedia Packet Transport : Multiple Layers or Descriptions?,” in *Packet Video'03*, Avril 2003.
- [Goy01] V.K. GOYAL, “Multiple Description Coding : Compression Meets the Network,” *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 74–93, September 2001.
- [Gué01] JEAN-PIERRE GUÉDON, *L'Histoire de la Transformée Mojette (la Forme et la Projection)*, Habilitation à Diriger des Recherches, Université de Nantes, 2001.
- [Kat78] MYRON KATZ, “Questions of uniqueness and resolution in reconstruction from projection,” in *Lecture Notes in Biomathematics*, vol. 26. Springer, December 1978.
- [Par01] BENOÎT PARREIN, *Description Multiple de l'Information par Transformation Mojette*, Thèse de doctorat, Université de Nantes et Ecole Centrale de Nantes, Novembre 2001.
- [PNG03] BENOÎT PARREIN, NICOLAS NORMAND, AND JEAN-PIERRE GUÉDON, “Multimedia Forward Error Correcting Codes,” *Annals of Telecommunications*, vol. 58, no. 3-4, pp. 448–463, April 2003.

## SynDEx et Méthodologie AAA

- [Gra00] T. GRANDPIERRE, *Modélisation d'architectures parallèles hétérogènes pour la génération automatique d'exécutifs distribués temps réel optimisés*, Ph.D. thesis, Université de Paris Sud, Spécialité électronique, novembre 2000.
- [LS92] C. LAVARENNE AND Y. SOREL, “Specification, performance optimization and executive generation for real-time embedded multiprocessor application with SynDEx,” in *Real Time Embedded Processing for Space Applications, CNES International Symposium*, November 1992.
- [LS97] C. LAVARENNE AND Y. SOREL, “Modèle unifié pour la conception conjointe logiciel-matériel,” *Traitement du Signal*, pp. 6–14, 1997.
- [Pra86] V. PRATT, “Modeling concurrency with partial orders,” *International Journal of Parallel Programming*, pp. 1–15, 1986.
- [RUN<sup>+</sup>05] MICKAËL RAULET, FABRICE URBAN, JEAN-FRANÇOIS NEZAN, CHRISTOPHE MOY, OLIVIER DÉFORGES, AND YVES SOREL, “Rapid Prototyping for Heterogeneous Multicomponent Systems : an MPEG-4 Stream over an UMTS Communication Links,” A paraître dans *EURASIP Journal on Applied Signal Processing*, 2005.
- [Sei00] R. SEINDAL, “GNU M4,” version 1.4, [www.seindal.dk/rene/gnu/man/m4\\_toc.html](http://www.seindal.dk/rene/gnu/man/m4_toc.html), 2000.
- [Tis03] ARNAUD TISSERAND, “DSP : des processeurs dédiés au traitement numérique du signal,” in *Séminaire du LIP*, Mai 2003.

## Estimation du mouvement

- [BB82] DANA H. BALLARD AND CHRISTOPHER M. BROWN, *Computer Vision*, Prentice-Hall, Inc, New Jersey, 1982.

- [Ber99] CHRITOPHE BERNARD, *Ondelettes et problèmes mal posés : la mesure du flot optique et l'interpolation irrégulière*, Ph.D. thesis, Ecole Polytechnique, 1999.
- [Bes03] NICHOLAS BESER, *Image Compression and Packet Video - Lectures 4, 5 and 6*, Programs in Engineering and Applied Sciences, John Hopkins University, Writing School of Engineering - Applied Physics Laboratory, September 2003.
- [Bru01] ERIC BRUNO, *De l'estimation locale à l'estimation globale de mouvement dans les séquences d'images*, Ph.D. thesis, Université Joseph Fourier, Grenoble - Laboratoire des Images et Signaux, Novembre 2001.
- [CCM<sup>+</sup>97] SHIH-FU CHANG, WILLIAM CHEN, HORACE J. MENG, HARI SUNDARAM, AND DI ZHONG, "VideoQ : an Automated Content Based Video Search System using Visual Cues," in *Proceedings of the fifth ACM international conference on Multimedia*, 1997, pp. 313–324.
- [CLL95] FRANÇOIS CHAROT, CLAUDE LABIT, AND PASCAL LEMONNIER, "Etude Architecturale d'un Algorithme d'Estimation de Mouvement Bloc-Récurif," Tech. Rep. 919, IRISA, Equipe Temis, Rennes, Mars 1995.
- [CM04] DUBHE CHAVIRA-MARTÍNEZ, *Schéma de codage vidéo orienté-texture : une approche robuste et scalable pour la diffusion sur réseaux*, Ph.D. thesis, IRISA, équipe TEMICS, Rennes, 12 Juillet 2004.
- [CS04] DANIEL CREMERS AND STEFANO SOATTO, "Motion Competition : a Variational Approach to Piecewise Parametric Motion Segmentation," *International Journal of Computer Vision*, 2004, A paraître.
- [Deb00] GILLES DEBUNNE, *Animation multirésolution d'objets déformables en temps-réel, Application à la simulation chirurgicale*, Ph.D. thesis, Institut National Polytechnique de Grenoble, Dec 2000.
- [Gor99] SEYED MOHAMMAD MEHDI MOSTAFAVI GORJI, *Une Méthode de Segmentation Spatio-Temporelle Opérant par Analyse Globale du Mouvement*, Ph.D. thesis, INSA de Rennes, Décembre 1999.
- [Hem01] MAGNUS HEMMENDORFF, *Motion Estimation and Compensation in Medical Imaging*, Ph.D. thesis, Institute of Technology, Department of Biomedical Engineering, Linköpings Universitet, Sweden, Juillet 2001.
- [HS81] B.K.P. HORN AND B.G. SCHUNCK, "Determining Optical Flow," *AI Memo 572*, vol. 17, no. 1-3, pp. 185–203, August 1981.
- [HS93] B.K.P. HORN AND B.G. SCHUNCK, "Determining Optical Flow : A Retrospective," *Artificial Intelligence*, vol. 59, pp. 81–87, 1993.
- [Kon00] JANUSZ KONRAD, "Motion Detection and Estimation," in *Handbook of Image and Video Processing*, A. Bovik, Ed., chapter 3.10, pp. 207–225. Academic Press, 2000.
- [Man03] ANTOINE MANZANERA, *Analyse d'images et vision : Estimation du mouvement dans les séquences d'images*, ENSTA / LEI, 2003.
- [Mar98] XAVIER MARICHAL, *Motion Estimation and Compensation for Very Low Bitrate Video Coding*, Ph.D. thesis, Université catholique de Louvain - Laboratoire de Télécommunications et Télédétection, Mai 1998.

- [Mar00] GWENAËLLE MARQUANT, *Représentation par Maillage Adaptatif Déformable pour la Manipulation et la Communication d'Objets Video*, Ph.D. thesis, IRISA/INRIA - Equipe Temics, Rennes, Décembre 2000.
- [MM99] JOSEP RAMON MORROS AND FERRAN MARQUÉS, "A proposal for dependent optimization in scalable region-based coding systems," in *IEEE International Conference on Image Processing*, Octobre 1999, vol. 4, pp. 295 – 299.
- [Mor98] FRANCK MORIER, *Méthodes de représentation hiérarchique du contenu des séquences d'images animées*, Ph.D. thesis, Laboratoire Systèmes Electroniques et Informatiques - IRESTE Nantes, Septembre 1998.
- [MRM03] GRÉGOIRE MERCIER, CHRISTIAN ROUX, AND GILBERT MARTINEAU, *Technologies du Multimédia*, ENST Bretagne, Janvier 2003.
- [Nzo95] VICTOR NZOMIGNI, *Compression sans perte de séquences d'images biomédicales*, Ph.D. thesis, Université de Rennes 1, Décembre 1995.
- [OB95] J.M. ODOBEZ AND P. BOUTHEMY, "Robust multiresolution estimation of parametric motion models," *Journal of Visual Communication and Image Representation*, vol. 6, no. 4, pp. 348–365, Décembre 1995.
- [OB98] J.M. ODOBEZ AND P. BOUTHEMY, "Direct incremental model-based image motion segmentation analysis for video analysis," *Signal Processing*, vol. 66, pp. 143–155, 1998.
- [Odo94] JEAN-MARC ODOBEZ, *Estimation, détection et segmentation du mouvement : une approche robuste et markovienne*, Ph.D. thesis, Université de Rennes 1, IRISA No 1304, Décembre 1994.
- [PBK94] M. PETROU, M. BOBER, AND J.V. KITTLER, "Multiresolution motion segmentation," in *International Conference on Pattern Recognition, Conference A : Computer Vision and Image Processing*, Octobre 1994, vol. 1, pp. 379–383.
- [RUGBRSH02] J.A. RODRIGUEZ, C. URDIALES GARCIA, A. BANDERA RUBIO, AND F. SANDOVAL HERNANDEZ, "A multiresolution spatiotemporal motion segmentation technique for video sequences based on pyramidal structures," *Pattern Recognition Letters*, vol. 23, no. 14, pp. 1761–1769, December 2002.
- [SK99] CHRITOPH STILLER AND JANUSZ KONRAD, "Estimating Motion in Image Sequences," *IEEE Signal Processing Magazine*, pp. 70–91, July 1999.
- [SM99] P. SALEMBIER AND F. MARQUES, "Region-based representations of image and video : Segmentation Tools for Multimedia Services," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1147–1169, December 1999.
- [STMG95] PHILIPPE SALEMBIER, LUIS TORRES, FERNAND MEYER, AND CHUANG GU, "Region-based video coding using mathematical morphology," *Proceedings of the IEEE*, vol. 83, no. 6, pp. 843 – 857, Juin 1995.
- [Tan03] STEPHANUS SURIJADARMA TANDJUNG, *Multiresolution Motion Estimation using Markov Random Field Modeling*, Ph.D. thesis, Nanyang Technological University, Parallel Distributed Computing Centre, 2003.

- [WD02] JULIANA FERNANDES CAMAPUM WANDERLEY AND DANIEL C. DANNA, “Temporal segmentation of video sequences for content-based coding,” in *Proceedings of Brazilian Symposium on Computer Graphics and Image Processing - SIBGRAPI 2002*, Fortaleza, CE, Brazil, October 2002.
- [Wu04] YING WU, *Optical Flow and Motion Analysis*, Electrical and Computer Engineering - Northwestern University, Advanced Computer Vision Notes Series 6, Spring Quarter 2004.
- [YM02] YUFEI YUAN AND MRINAL K. MANDAL, “Low-Band-Shifted Hierarchical Backward Motion Estimation and Compensation for Wavelet-Based Video Coding,” in *3rd Indian Conference on Computer Vision, Graphics and Image Processing*, Decembre 2002, pp. 185–190.
- [YR00] XUGUANG YAND AND KANNAN RAMCHANDRAN, “Sacalable Wavelet Video Coding using Aliasing-Reduced Hierarchical Motion Compensation,” *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 778–791, Mai 2000.
- [YRA97] SEUNG CHUL YOON, KRISHNA RATAKONDA, AND NARENDRA AHUJA, “Region-based video coding using a multiscale image segmentation,” in *International Conference on Image Processing, ICIP’97*, Octobre 1997, vol. 2, pp. 510–513.
- [YRA99] S.C. YOON, K. RATAKONDA, AND N. AHUJA, “Low Bit-Rate Video Coding with Implicit Multiscale Segmentation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 7, pp. 1115–1129, October 1999.
- [ZBK97] KUI ZHANG, MIROSLAW BOBER, AND JOSEF KITTLER, “Image sequence coding using multiple-level segmentation and affine motion estimation,” *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 9, pp. 1704–1713, Septembre 1997.
- [ZC97] DI ZHONG AND SHIH-FU CHANG, “Spatio-Temporal Video Search using the Object Based Video Representation,” in *International Conference on Image Processing*, Octobre 1997, vol. 1, pp. 21–24.
- [ZL01] DENGSHENG ZHANG ZHANG AND GUOJUN LU, “Segmentation of moving object in image sequence : A review,” *Circuits, Systems, and Signal Processing (Special Issue on Multimedia Communication Services)*, vol. 20, no. 2, pp. 143–183, 2001.

## Perspectives : techniques envisagées

- [CB03] PATRICK LE CALLET AND DOMINIQUE BARBA, “Robust approach for color image quality assessment,” in *Visual Communications and Image Processing, VCIP’03*, 2003, vol. 5150 of *Proceedings of SPIE*, pp. 1573–1581.
- [CSE05] A. CAVALLARO, O. STEIGER, AND T. EBRAHIMI, “Tracking video objects in cluttered background,” *IEEE Trans. Circuits and Systems for Video technology*, vol. 15, no. 4, pp. 575–584, April 2005.